Sheridan College

SOURCE: Sheridan Institutional Repository

Publications and Scholarship Faculty of Applied Science & Technology (FAST)

12-2018

Reasoning about ideal interruptible moments: A soft computing implementation of an interruption classifier in free-form task environments

Edward R. Sykes
Sheridan College, ed.sykes@sheridancollege.ca

Follow this and additional works at: https://source.sheridancollege.ca/fast_publications

Part of the Computer Engineering Commons

Let us know how access to this document benefits you

SOURCE Citation

Sykes, Edward R., "Reasoning about ideal interruptible moments: A soft computing implementation of an interruption classifier in free-form task environments" (2018). *Publications and Scholarship*. 43.
https://source.sheridancollege.ca/fast_publications/43

This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License.
This Article is brought to you for free and open access by the Faculty of Applied Science & Technology (FAST) at SOURCE: Sheridan Institutional Repository. It has been accepted for inclusion in Publications and Scholarship by an authorized administrator of SOURCE: Sheridan Institutional Repository. For more information, please contact source@sheridancollege.ca.

Contents lists available at ScienceDirect

International Journal of Human-Computer Studies

journal homepage: www.elsevier.com/locate/ijhcs

Reasoning about ideal interruptible moments: A soft computing implementation of an interruption classifier in free-form task environments

Edward R. Sykes

Sheridan College, School of Applied Computing, 1430 Trafalgar Rd, Oakville, ON L6H 2L1, Canada

ARTICLE INFO

Keywords:
Human computer interaction
Interruption
Machine learning
Soft computing applications
Situationally appropriate interaction
Workload

ABSTRACT

Current trends in society and technology make the concept of interruption a central human computer interaction problem. In this work, a novel soft computing implementation for an Interruption Classifier was designed, developed and evaluated that draws from a user model and real-time observations of the user's actions as s/he works on computer-based tasks to determine ideal times to interact with the user. This research is timely as the number of interruptions people experience daily has grown considerably over the last decade. Thus, systems are needed to manage interruptions by reasoning about ideal timings of interactions.

This research shows: (1) the classifier incorporates a user model in its' reasoning process. Most of the research in this area has focused on task-based contextual information when designing systems that reason about interruptions; (2) the classifier performed at 96% accuracy in experimental test scenarios and significantly outperformed other comparable systems; (3) the classifier is implemented using an advanced machine learning technology—an Adaptive Neural-Fuzzy Inference System—this is unique since all other systems use Bayesian Networks or other machine learning tools; (4) the classifier does not require any direct user involvement—in other systems, users must provide interruption annotations while reviewing video sessions so the system can learn; and (5) a promising direction for reasoning about interruptions for free-form tasks–this is largely an unsolved problem.

1. Introduction

Determining when to interrupt a user at appropriate times as s/he performs computer-based tasks is an ongoing problem (Altmann et al., 2014; Baethge et al., 2014; Iqbal and Bailey, 2010). From an algorithmic perspective, it is difficult to determine the precise time to interrupt a user. This is because there are several subproblems that need to be solved to be confident that an interruption will be beneficial to the user. Some subproblems include: i) determining the intent (or goal) of the user as s/he is performing the task; ii) determining the task difficulty (Gievska et al., 2005; Gievska and Sibert, 2004); iii) determining the user's current cognitive load (Gievska et al., 2005; Iqbal and Bailey, 2006); iv) estimating the cost of the interruption and the resumption lag time (Iqbal and Bailey, 2005); and v) incorporating personal user characteristics, such as sensitivity to being interrupted, distractibility level, etc. (Horvitz et al., 2003). A solution to these problems is needed to make accurate decisions about the timing of interruptions.

Since interruption is a key human-computer interaction problem, systems must be developed to manage interruptions in terms of reasoning about ideal timings of interruptions. In designing the classifier, the following desirable characteristics were identified:

1. make accurate decisions when uncertainty is present;
2. be computationally efficient and able to make interruption decisions in real time (however, the classifier learning does not need to be real time);
3. employ a user model (e.g., preferences, familiar tasks, etc.) that is used in the decision-making process;
4. draw on direct measurements from user activities;
5. learn from non-linear input data (i.e., assume input is not necessarily linear);
6. provide a reasoning process that is easily interpretable by a human (i.e., the classifier's decision-making process to interrupt or defer an interruption must be easy to be examined and understood by a human). This characteristic provides the opportunity for deeper reasoning into why an interruption occurred as well as insight into the form and content of an appropriate message for user interaction;
7. support supervised learning (capable of accepting input-output patterns and learning these associations); and
8. learn quickly from a small number of training data sets.

E-mail address: ed.sykes@sheridanc.on.ca.

https://doi.org/10.1016/j.ijhcs.2018.06.005
Received 9 December 2016; Received in revised form 20 June 2018; Accepted 22 June 2018
Available online 27 July 2018
1071-5819/ © 2018 The Author. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license
(http://creativecommons.org/licenses/BY-NC-ND/4.0/).

1.1. Outcomes and contributions

We created a machine learning classifier that performs as well as user-determined interruption points. The rationale why user-determined interruption timings are ideal is explained in the following sections. Additional outcomes and contributions include:

1. The classifier incorporates a user model in its' reasoning process. Our classifier includes both user and task contextual information—other classifiers include task details only.
2. In the best models constructed, our classifier performs at an accuracy of 98% with historic event knowledge. This level of performance exceeds comparable studies in this area of HCI (Finn and Limerick, 2003; Lisetti and Nasoz, 2004; Picard, 2003; Pu et al., 2006).[1]
3. The classifier was implemented using an advanced machine learning technology (i.e., Adaptive Neuro-Fuzzy Inference System)—which is a novel contribution.
4. This research expands our understanding on reasoning about ideal interruption points for free-form tasks. Currently, this is largely an unsolved problem.
5. The classifier was designed as a framework so it could be generalized to other tasks and problem domains.

The structure of this paper is: Section 2 presents a literature review of interruption research and a survey of candidate machine learning algorithms. Section 3 presents the design and methodology including the classifier requirements, the Interruption Classifier, data capturing techniques, and the details of the machine learning technologies used in the implementation of the classifier. Section 4 presents the findings (analysis and evaluation), Section 5 presents a discussion on the implications of the empirical results for theories of interruption and implications for deploying classifiers for detecting interruptible moments in practice. Lastly, Section 6 presents the conclusions.

2. Literature review

The goal of this research is to identify the most appropriate classifier for predicting the interruptible moment given the context and task of the user. As previously discussed, the classifier created draws information from a user model and real-time data of the users' actions. Acknowledging that the literature in this area is very broad and diverse, the review conducted specifically concentrated on the desirable characteristics as outlined above. For example, criterions 1 and 5–8 reduce the number of machine learning algorithms suitable for review.

2.1. Interruption

Interruptions happen for a multitude of reasons and there are four known strategies for managing them: (a) immediate, (b) scheduled, (c) negotiated, and (d) mediated (Guinn, 1999; McFarlane and Latorella, 2002). The *immediate* interruption strategy involves interrupting the person immediately regardless of what they are doing in a way that insists that the user immediately stop what they are currently working on and respond to the interruption. The *scheduled* strategy involves restricting the agents' interruptions to a prearranged schedule. The *negotiated* interruption strategy would have the agent announce their need to interrupt and then support a negotiation with the person. This approach gives the user full control over how to deal with the interruption—when or even at all. The fourth strategy, called *mediated*, involves agents indirectly interrupting and requesting interaction through a broker like a smartphone. The smartphone would then determine when and how the agents would be allowed to interrupt the user. The Interruption Classifier is designed as a broker with the intelligence to reason about when to interrupt the user.

Most of the current research is focused on mediated and negotiated strategies with research in the mediated strategy area growing considerably (Altmann et al., 2014; Baethge et al., 2014; Iqbal and Bailey, 2010). Associated with mediated strategies are intelligent systems that observe the user as s/he is performing tasks to decide when to interrupt the user and how best to present the pertinent information. Despite the progress that has been made in systems supporting mediated strategies, negotiated strategies (user determined) are still the best overall solution when considering factors such as cost of interruption, resumption lag, and overall performance in carrying out multiple tasks (McFarlane and Latorella, 2002). The proposed system corresponds to the 'mediated' strategy.

The following section presents these topics on interruptions: (a) tasks and task boundaries; (b) cognitive load, cost of interruption and resumption lag; and (c) models of interruption and an interruption taxonomy.

2.1.1. Tasks and task boundaries

Task and interruption researchers are interested in acquiring contextual information surrounding the task so that the timing of the interruption and the information presented will be minimally disruptive and of the utmost benefit to the user at that time (Iqbal and Bailey, 2007). Reasoning systems must incorporate task properties because these systems must be able to decide optimal times to interrupt. This decision often hangs on the very task the user is engaged in at the time (Iqbal and Bailey, 2007). In many situations if it is possible to defer an interruption to a task boundary, the inconvenience to the user by responding to the interruption is significantly reduced (Iqbal and Bailey, 2007). In these situations the resumption lag is much less for the user than if the interruption occurred during the task (Altmann et al., 2014; Baethge et al., 2014). The concept of a task boundary will be integrated into the classifier as one of the features that it implicitly learns through user training data sets.

2.1.2. Automatic task boundary identification

Task boundary identification techniques are used to detect and identify breakpoints during tasks to establish policies for interruption software (Iqbal and Bailey, 2007). Researchers have focused on building statistical models that dynamically extract characteristics of the interaction to a specific type of breakpoint (i.e., coarse, medium, fine) (Horvitz and Oliver, 2005; Iqbal and Bailey, 2007). The findings indicate that these models can pick out task breakpoints with a reasonable amount of accuracy for prescribed tasks (Iqbal and Bailey, 2007). An example of a prescribed task is solving a jig-saw puzzle. Each individual subtask is the act of dragging a piece to its appropriate location in the jig-saw puzzle.

As it relates to interruption, these models could be used to augment interruption management software to effectively determine better times to interrupt the user by establishing defer-to-breakpoint policies (Iqbal and Bailey, 2007). However, these studies have primarily focused on prescribed tasks. It is significantly more difficult to detect breakpoints within tasks that are highly variable in nature. For example, free-form tasks are by far the most common type of computer-based task and are still largely an unsolved problem for interruption researchers (Gluck et al., 2007; Horvitz et al., 2004; Iqbal and Bailey, 2005; Iqbal and Bailey, 2008). In the context of this research the definition of a *free-form task* is one in which the tasks are highly variable and the interaction including actions and timing between the user and the task cannot be predicted.

Currently there are very few algorithms that pick ideal times to interrupt a person working on free-form tasks (Please see: (Fogarty et al., 2005; Iqbal and Bailey, 2008)). This is one area where this research contributes by providing an innovative solution to this difficult

[1] A discussion regarding levels of performance for machine learning classifiers is further elaborated in Section 5.

problem.

2.1.3. Cognitive load, cost of interruption and resumption lag

Cognitive load is an indicator of the degree of working memory utilized when the user is performing a task (Hertzum and Holmegaard, 2013).

The *Cost of Interruption* (COI) is a subjective measure of a user's wish to remain undisturbed while working on a computer based task (Hertzum and Holmegaard, 2013). We acknowledge that other definitions of COI exist, however, in this research we use the previous definition. The COI may include various kinds of alerts disrupting a user in different contexts (Horvitz et al., 2003, 2004). The COI has been used as an assessment tool for several decades in decision analysis in various fields (Horvitz et al., 2004).

Resumption Lag (RL) is defined as the time required to resume the primary task after completing the interrupting task (Iqbal and Bailey, 2005). RL can be measured as the time from closing the interrupting task to the first keyboard or mouse action in the primary task in direction of the task goal (Iqbal and Bailey, 2005).

There is a strong correlation between cognitive load and the COI (Iqbal and Bailey, 2005). Thus, it is important to assess the cognitive load on the user while s/he is performing a task to decide whether to interrupt the user. Researchers have shown that if a user is interrupted during a high cognitive load task by being forced to switch tasks, then the COI can be very high (Iqbal and Bailey, 2005).

Consequently, an important design consideration for the Interruption Classifier was that it considers the user's workload or cognitive load when deciding whether to interrupt the user. A design aspect of our classifier acknowledges that the COI can be reduced by aligning the interruptible moment with subtask boundaries (Fogarty et al., 2005; Iqbal and Bailey, 2008). Furthermore, it should be noted that the identification of ideal interruption points for free-form tasks is largely an unsolved problem (Altmann et al., 2014; Baethge et al., 2014). Therefore, uncertainty is part of what this classifier needs to consider (criterion #1).

2.2. Ideal interruption points

In this research, an *ideal interruption point* is defined as the time when a user would normally choose to serve an interruption while considering: (1) the user's cognitive load should be low (Gievska et al., 2005; Iqbal and Bailey, 2006); (2) the user should be at a coarse task breakpoint (e.g., about to switch from a spreadsheet to email); and (3) the length of the interruption task should be short so that the user can quickly return to the primary task without the loss of continuity in performing the primary task.[2] Criterions 1 and 2 are design characteristics of our classifier; criterion 3 is supported by the experiments we have prepared. Our motivation for designing a system that learns from user determined timings stems from the literature that indicates that users *prefer* full control over when to serve interruptions (Altmann et al., 2014; Baethge et al., 2014; McFarlane and Latorella, 2002). The primary purpose of this research is *to design a system that serves interruptions at that same times that s/he would most prefer to be interrupted.* This research does not focus on designing a system that increases the overall performance of the user. If a real-world interruption management system is implemented using our classifier, it would suggest interruptions at times that would be most in tune with his/her interruption preferences. This would enable the system to serve as an effective *mediator*–one that would receive incoming interruption requests for the user and decide, based on that user's characteristics, when is the most appropriate time to interrupt him/her.

2.3. Models of interruption and an interruption taxonomy

Models of Interruption refer to the set of models researchers have proposed to assist in representing the context from which an interruptible moment may be reasoned about. The Memory for Goals model has been used by many different researchers to understand and model interrupted task performance for nearly two decades (Altmann and Trafton, 2002; Bower and Morrow, 1990). There have been a myriad of studies mostly based on the Memory for Goals model that provide support for why an interruption alert is useful and improves performance (Altmann and Trafton, 2002; Bower and Morrow, 1990). The Memory for Goals model helped the design of our experiment during which participants choose their own times to serve interruptions after an alert. Currently, each model is very specific to the types of tasks intended to be performed by its users. Much of the research to date has been centered on using attributes from the task domain (Altmann et al., 2014; Iqbal and Horvitz, 2010). However, there are other aspects surrounding the problem of determining when to interrupt a user (Loukopoulos et al., 2009). These other aspects involve the user and environment contexts. Models have been proposed to provide a more encompassing perspective of interruption; however, currently, there is no standard or unified model that has been accepted in the research community. Currently, there are several interruption taxonomies (Gievska and Sibert, 2005; McFarlane and Latorella, 2002). We focused on the one proposed by Gievska & Sibert since it aligns well with the goals of this research. This taxonomy includes three dimensions: (1) User Context captures the salient features of the user's characteristics and traits; (2) Environment Context represents attributes representing the user's current working environment; (3) Task Context aims to capture the significant properties of the computer based task (Fig. 1).

The purpose of this taxonomy is to serve as a framework to identify attributes and relationships appropriate for conceptualizing factors that influence the timing of interruption. Gievska & Sibert have designed and implemented a system to automatically detect interruption points; however, they have only designed systems that focused on Task Context variables—User and Environment context variables were not included (Gievska et al., 2005). “Subjective preferences were not considered as a factor in the current implementation of the interruption mediator. However, they should not be neglected when designing user interfaces that give equal priority to user's satisfaction and comfort as to other performance measures.” pg. 176, (Gievska et al., 2005). It follows then that the direction of research is to embrace aspects from all three dimensions.

2.4. Machine learning algorithms

Before designing the Interruption Classifier, many machine learning algorithms were reviewed (i.e., Bayesian networks, neural nets, statistical classifiers, etc.). Many of these were discarded because they failed to meet one or more of the desirable characteristics and they are limited for modeling human behaviour. Certain machine learning techniques in the area of *Soft Computing* have been applied to a number of user modeling problems by learning from user behaviour and integrating them as part of the user model (Frías-Martínez et al., 2004). Soft computing is a family of methods that are based on fuzzy logic, neural networks, and probabilistic reasoning tools (Jang et al., 1997). Furthermore, soft computing algorithms have been shown to be very effective at deriving solutions to problems where other approaches have

[2] Although there is no consensus in the research community on what constitutes an interruption task that is short, for this research, it is defined as a task that is 10s or less. This definition has support from the following references: (Hodgetts & Jones, 2003, 2007; Solingen, Berghout, & Latum, 1998). Although there are many different definitions to *interruption*, for this research, the definition used is provided at the beginning of Section 2. This means that we do not take into account interruptions that are integrated in the task (e.g., help systems). All the interruptions are on issues/tasks outside of the primary task. As discussed in Section 3, this experiment focuses on two unrelated tasks (a primary task and an interruption task) and they have been designed to be unrelated.

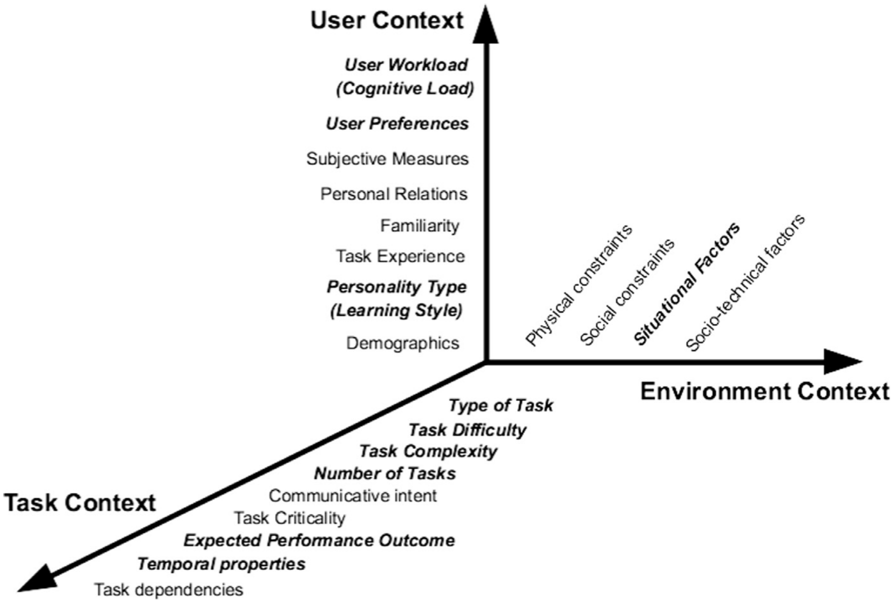


Fig. 1. Interruption taxonomy (Gievska et al., 2005).

Table 1
Soft computing characteristics applied to user modeling (Adapted from: Frias-Martinez, et al., 2004).

	Complexity	Dynamic modeling	Size of training data	Uncertainty	Noisy data	Interpretability
Fuzzy inference systems	✓	✓✓✓	—	✓✓✓	✓✓✓	✓✓✓
Neural networks	✓✓	✓✓✓	✗	✓✓✓	✓	✗✗
Neuro-Fuzzy	✓✓✓	✓✓✓	✓✓✓	✓✓✓	✓✓✓	✓✓✓

Legend: ✓✓✓: very suitable, ✓: suitable, ✗: unsuitable, ✗✗✗: very unsuitable, —: Not applicable.

failed (Jang et al., 1997). Table 1 summarizes the characteristics of different soft computing techniques based on six criteria from Frias-Martinez et al. (2004):

1. Computational complexity: ability for the results to be computed within 1s in current computing environments (the decision to interrupt now or defer interruption needs to be real-time);
2. Dynamic modeling: the ability to adapt (or change) based on the user model on-the-fly;
3. Size of training data: the amount of data needed to produce a reliable user model;
4. Uncertainty: the ability of the techniques to handle uncertainty;
5. Noisy data: the ability to handle noisy data (i.e., how noisy training data will affect the user model);
6. Interpretability: the ease with which a human can interpret the results of the knowledge captured.

Human interaction is a key component of any user modelling application—like the Interruption Classifier—which implies that the data available will be most likely imprecise, incomplete and heterogeneous (Frías-Martínez et al., 2004). In this context Soft Computing, specifically Neuro-Fuzzy systems appear to be the appropriate paradigm to handle the uncertainty in this problem. Adaptive Neuro-Fuzzy Inference Systems (ANFIS) offer these benefits:

1. domain expert's knowledge can be embodied in a Fuzzy Inference System (FIS) by extracting and describing the knowledge using linguistic variables and membership functions—an ANFIS can use this FIS as an initial design for modeling the problem domain (Negnevitsky, 2004);
2. Faster convergence than typical feedforward neural networks (Garcıa and Mendez, 2007);
3. ANFIS requires a smaller size training set to converge when compared to other machine learning tools (Gharaviri et al., 2008);
4. Smoothness is guaranteed by the fuzzy logic inference mechanism (Negnevitsky, 2004);
5. Automatic fuzzy logic parametric tuning (Jang et al., 1997);
6. Easily inspectable and interpretable by humans (Frías-Martínez et al., 2004);
7. One of the top choices for user modelling based on user modeling researchers (Frías-Martínez et al., 2004).
8. Wide-level of acceptance from academics to industry specialists. ANFIS have been used to solve a variety of academic research problems and applied industrial problems.

The selection and use of an ANFIS for this research appears to have great promise since it satisfies all the desired characteristics; offers a significant amount of flexibility because it fits the soft computing paradigm; and has long-standing support from the user modeling community as a viable tool for user modeling problems (Frías-Martínez et al., 2004; Jang et al., 1997). As a result, an ANFIS was chosen as the machine learning tool for the Interruption Classifier. In summary, the goal of this research is to design a classifier that will perform as well as user-determined interruption timings as outlined in Section 1.

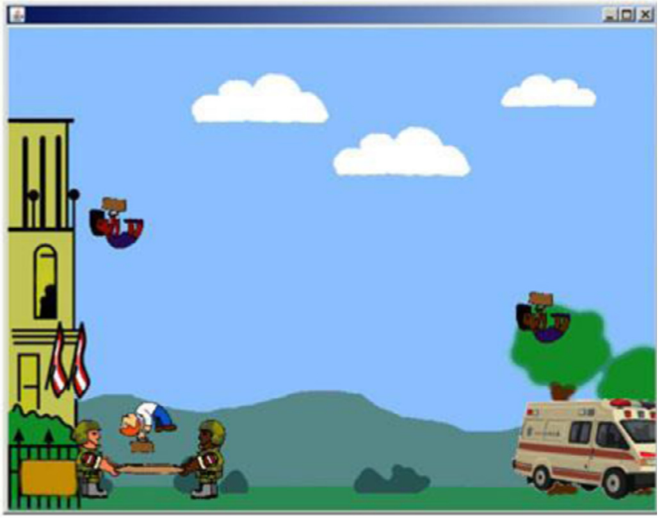


Fig. 2. Primary task: stretcher-bearers bounce characters jumping out of a building to the waiting ambulance.

3. Methodology

This section presents the details of the methodology. It includes the approach taken to identify appropriate computer-based tasks, machine learning techniques, and the experiment and analysis techniques. Additionally, the design of the Interruption Classifier is presented. The following section describes various aspects of the methodology including: experiment task design; pilot studies; Interruption Classifier; experiment procedure; participants; methodology for designing, training, testing and refining the Interruption Classifier; and analysis techniques.

3.1. Experiment task design

The experiment design involved a primary task (Jumping game) and an interruption task (intermittent Matching game). The primary task is modeled after a video game by Nintendo called *Fire* and McFarlane's interruption work (McFarlane and Latorella, 2002). The interruption task is modeled after the matching tasks used in experiments of the Stroop Effect (Tulga and Sheridan, 1980). The tasks are unrelated by design to ensure that different cognitive resources are required to complete each task. The experiment task design is largely based on McFarlane and Latorella (2002).

3.1.1. The primary task

This task requires the user to move stretcher-bearers to catch other game characters as they jump from a building. Fig. 2 depicts the game in which the user must successfully bounce each falling character three times in three different locations and into the awaiting ambulance. If a character lands on the ground, then that character is not saved. The game is simple when only one character at a time jumps out of the building; however, when many game characters are jumping at a time, the game becomes more difficult. A *subtask* is defined as the task for the participant to manage an individual jumping character and to save him/her. This is accomplished by moving the stretcher in the appropriate position to ensure the character is bounced several times and then finally into the ambulance. The time between subtasks is sufficient such that the participant is not required to provide constant attention. This design is intended to provide participants the ability to successfully serve an interruption task if needed.

The game runs continuously, so even if the participant serves an interruption task, the game continues. The composition of the game permits observation of participants' behaviours to be directly mapped onto discrete subtasks. The primary task offers the following beneficial task characteristics:

- The participant's performance on successfully completing a subtask can be classified as either true or false.
- Subtasks require participants to make time-sensitive decisions.
- Subtasks do not require persistent attention from participants.
- Participants need to be aware of the state of each jumper to be successful in completing each subtask. Thus, there is an overhead cost for resuming the primary task after serving an interruption task.

The primary task was designed to enable analysis of ideal and undesirable interruption points. The primary task was also designed to be uncomplicated so that additional noise would be reduced as much as possible:

- Each subtask takes 16.9 s from the start of the jump until it lands safely in the ambulance. (The time from its initial jump to its third (and last) bounce is 13.7 s and the time for its third bounce into the ambulance is 3.2 s).[3] Please note the starting time when a jumper jumps is completely unpredictable by design.
- Subtasks are completely independent. An error on one subtask does not impose errors on other subtasks.
- The primary task is easily learned in terms of the high-level goal to save jumpers and in controlling the stretcher-bearers—there are only two keyboard keys used for control.

3.1.2. The interruption task

A matching task was used as the interruption task. This task uses a graphical matching task that is based on Stroop effect studies (Jensen and Rohwer, 1966). This task requires the participant to make matching decisions (by shape or colour) based on the rule presented on the screen (Fig. 3) using these principles:

- Each problem in the matching task requires a definite but minimal focus of attention (a distinct amount of cognitive workload is required to resolve the conflict thus serving as a valid interruption task).
- The individual matching tasks are independent and the task cannot be automated through *overlearning* or *automate*d by repetitively performing the task—from one matching problem to the next each individual matching task may be viewed independently as a novel problem (Tulga and Sheridan, 1980).
- The amount of time required to complete one matching task is relatively consistent. This consistency allowed participants to plan their strategies to perform the primary and interruption tasks to the best of their ability.
- Matching tasks were performed one at a time from a queue (QueueSize: number of waiting interruption tasks)
- Interruptions of interruptions are not allowed (there were no interruption notifications issued while the participant was working on an interruption task).
- The same randomization scheme that was used to schedule the subtasks of the primary task was also used to schedule the individual interruption tasks.

The experiment was designed so that it is not possible to predict when an interruption task will appear, nor is it possible to predict how or when the user will interact with the interruption task. It follows then, from the definition of a free-form task presented in Section 2, that this

[3] These timings are a result of the design and implementation in relation to how the game and matching tasks were programmed. Details regarding ideal interruption points is presented in Section 5.

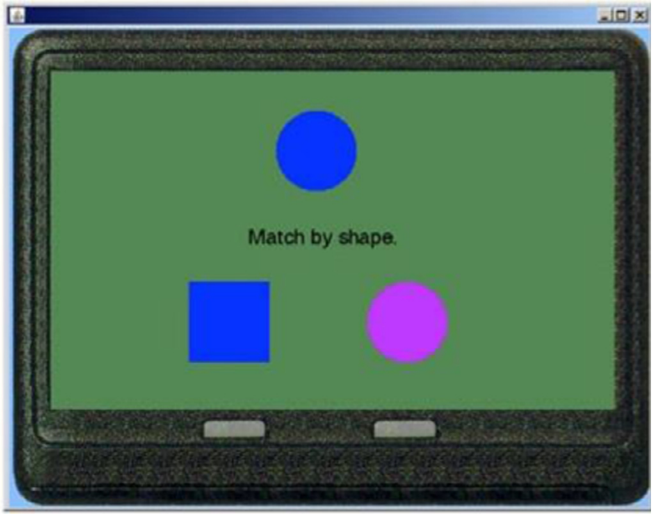


Fig. 3. Interruption task: *Match by colour* or *Match by shape*—imposes a level of cognitive load to resolve the conflict.

task is also a free-form task. Furthermore, viewed at a more abstract perspective, the combination of the primary and interruption tasks represents a free-form task.[4]

3.2. Full experiment

This section presents an overview of the experiment including treatments, objective and subjective measures, and data for the classifier. In preparing for the full experiment, a pilot study was conducted which represented all aspects of the whole experiment—from introduction to debriefing. The pilot study provided an opportunity to set the primary and interruption task complexity to the appropriate level; identify issues so that the experiment would run consistently and the results would be as reliable as possible; and to gain an understanding of the strategies and tactics relating to each participant's negotiated interruption style. Amongst the most interesting findings from the pilot study were anecdotes from the participants revealing their strategies:

“The matching task was demanding because you struggle to keep focus. I worked hard to keep focused during the matching task. It wasn't frustrating—it was fun! The majority of the time I switched when the jumpers were at the highest point, which would give me time to complete the interruption.” (Pilot Study Participant #1). “My strategy was to move the stretcher to the next bounce position, and then serve the interruption immediately. This gave me plenty of time to do the interruption during a full bounce cycle.” (Pilot Study Participant #3).

3.2.1. Treatments

Three treatment conditions were used in this research experiment.

1. Treatment #1: *Primary Task only* base case—no interruption tasks.
2. Treatment #2: *Interruption Task only* base case—no primary task.
3. Treatment #3: *Negotiated* treatment condition gave the participant full control over when s/he served the interruption task. When an interruption task was queued, the screen *blinked* for a moment on top of the primary task. The participant was then able to serve the interruption task at a time of his/her choice.

All the participants received the three treatments. To avoid potential confounds such as tiredness, each treatment was administered by two sequential trials with a 30 s rest period in between. Furthermore, a diagram-balanced Latin squares ordering was used for counterbalancing. For example, consider the following sequence of treatments:

Practice Session Data Trial Sequence: 3 3 1 1 2 2 (total: ~30min),
Experimental Session Data Trial Sequence: 3 3 1 1 2 2 (total: ~30min).

Table 2
Objective measures and collection properties.

	Data collected	Data capturing technique
1	Number of jumpers saved on the game task	Raw data count
2	Number of matches done wrong of those attempted	Raw data count
3	Number of key presses per jumper saved on game task	Raw data count
4	Average time in seconds from the scheduled onset of each interruption task until it was completed or the trial timed out.	Computed value from raw data count
5	Average time in seconds from display of each matching task until it was completed	Computed value from raw data count

This section describes the measures collected in the experiment and those used by the Interruption Classifier. For the experiment, the participant's performance was the dependent variable and was determined by the analysis of five objective measures and 17 subjective measures. For the classifier, the same objective measures were used with an additional set of measures for training and testing purposes.

3.2.2. Objective experiment measures

Table 2 depicts summative objective measures and data collection properties. The objective measures are grouped into the following categories: correctness (metrics 1 and 2), efficiency (metric 3), and timeliness (metrics 4 and 5). The purpose of collecting these measures was to be able to calculate the participant's performance with respect to the base cases and the negotiated interruption strategy. These measures were collected during trials with summative values computed at the end. However, during the experiment formative updates of these measures were computed. For example, the state of the number of jumpers saved, the number of matches done wrong, etc. was computed every millisecond within each trial (4 ½ min [270,000 ms]). Thus, there are approximately 270,000 data points (i.e., cases) for each trial containing all the contextual information for the classifier to reason about interruption points.

3.2.3. Subjective experiment measures

Two questionnaires were administered to the participants. An opening questionnaire captured the participant's characteristics (please see Appendix B). These measures collectively represent the core of the user model from which the classifier draws user-specific contextual information:

(a) participant's age (0–90);
(b) participant's comfort level with computers (general computer familiarity) (1–5, 5 = very familiar);
(c) familiarity with video games (1–5, 5 = very familiar);
(d) tolerance to being interrupted (1–5, 5 = don't mind interruptions);
(e) frustration level (1–5, 5 = easily frustrated);
(f) distractibility level (1–5, 5 = easily distracted);
(g) thinking style (1–5, 1 = thinking type of person, 5 = hands-on type of person);
(h) multi-tasking ability (1–5, 5 = very good at multi-tasking); and
(i) regaining focus after interruptions ability (1–5, 5 = difficult to regain focus).

These specific personal characteristics were measured because they represent subjective aspects that directly or indirectly impact how a

[4] A video excerpt of this HCI experiment is found here: https://youtu.be/05AQEZb8w_I

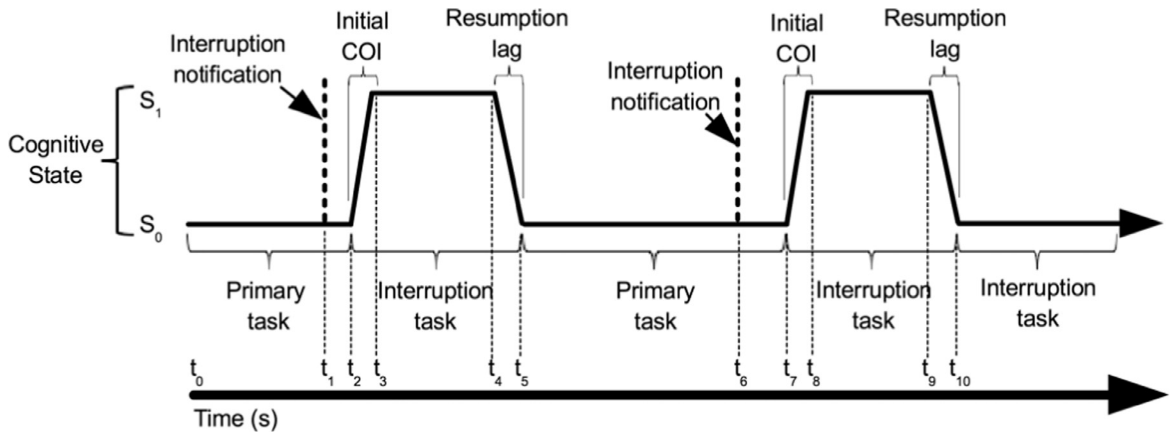


Fig. 4. Sample interruption timing scenario—alternating cognitive states ($S_0 \Leftrightarrow S_1$) and task states (primary ⇔ interruption tasks). At times t_2 and t_7 the participant decided to engage in performing the interruption task.

person perceives and responds to interruptions. One objective in this research involved exploring which of these personal characteristics were predictive of the interruption points. This is elaborated in the Discussion. The closing questionnaire was used to assess the subjective measurements of the participant's opinions on various aspects of the experiment (please see Appendix C).

3.2.4. Classifier measures—data for the classifier

Data were collected and used by the classifier to design a solution for the mediated interruption strategy. Fig. 4 presents a scenario to motivate the selection of data used by the classifier. The sequence diagram shows a participant undergoing the negotiated interruption strategy treatment:

(a) the times for interruption notifications ($t_{1,}$ and t_6),
(b) the initial COI to serve the interruption (depicted in the time sections t_2 to t_3 and from t_7 to t_8).
(c) the resumption lag time (shown in the time sections t_4 to t_5 and from t_9 to t_{10}).

The time from t_1 to t_3 and from t_6 through t_8 represent the entire contextualization from initial notification of an interruption to the time when the participant is fully engaged in the interruption task. However, the most important timings are when the participant decided to switch from the primary task to the interruption task. To train the classifier, timings t_2 and t_7 are especially important since it is at these times that the participant decided to serve the interruption. These timings, in combination with user model and task details were important to train the classifier.

The following data were used in designing the classifier to satisfy the research goal of designing a system that performs as well as the negotiated interruption strategy at detecting ideal interruption points. The data are:

1. Wall clock time (milliseconds).
2. Real-time values of the objective measures as the participant performs the primary and interruption tasks.
3. Time when the interruption notification was issued to participant.
4. Time when participant switched from the primary task to perform the interruption task.
5. Number of jumping characters currently visible on the screen.
6. State of jumping characters (e.g., where in the cycle of *bouncing up* or *falling down*).
7. Position of the stretcher in the primary task (i.e., one of three possible positions).
8. Number of interruption tasks waiting to be served (QueueSize).
9. Current activity (primary task or interruption task).
10. Participant's user model information: comfort level working on computers, tolerance to being interrupted, frustration level, distractibility level, multitasking ability, etc.

3.3. Interruption classifier

This section describes, at an abstract level, the Interruption Classifier. The classifier needs the following information to determine appropriate times to interrupt the user (*U*):

v: vector representing *U*'s real-time activities (extracted in real-time from the software being used by *U*);
w: vector representing the static data for the user (user modeling information such as personality traits, frustration level, tolerance to interruptions, etc.);
x: vector representing task and environment specific information (task complexity, contextual information, etc.);

Table 3
High-level description of the interruption classifier.

```
Interruption_classifier ($\vec{t}$, $\vec{v}$, $\vec{w}$, $\vec{x}$, trained ANFIS, interruption_threshold_limit)
    $\vec{t}$ : temporal contextual information
    $\vec{v}$ : real-time user activity data // Input from the software User (U) is using
    $\vec{w}$ : static user data // Input extracted externally from the software U is using
    $\vec{x}$ : task and environment specific information
trained_ANFIS
interruption_threshold_limit ← range: [0,1] | 0 = do not interrupt, 1 = interrupt

Begin
interrupt ← 0 // default is defer an interruption
while (U is interacting with system) // run ANFIS with new inputs
ANFIS ← $\vec{t}$ // temporal specific data
ANFIS ← $\vec{v}$ // real-time data
ANFIS ← $\vec{w}$ // static user data
ANFIS ← $\vec{x}$ // task, environment
// compute interruptDecision: ← [0,1] | 0 = do not interrupt, 1 = interrupt
If (interruptDecision > interruption_threshold_limit) then interrupt U
end while
End
```

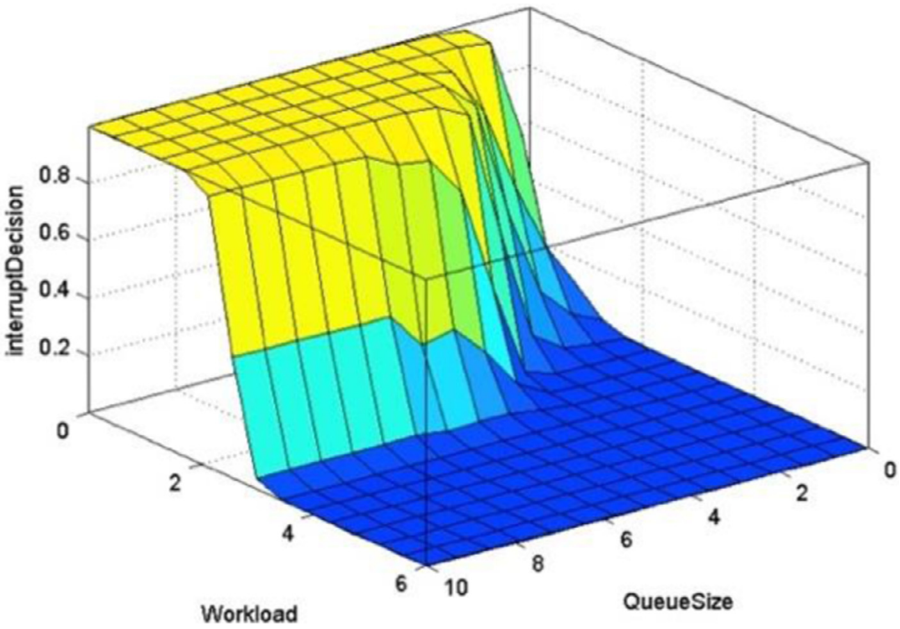


Fig. 5. Surface view of initial model depicting workload (x-axis) and QueueSize (y-axis) inputs with InterruptionDecision (z-axis) as the output variable (0 = Do Not Interrupt).

Table 4
Initial fuzzy rules for the design of the interruption classifier.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

The initial design of the classifier is based on a Fuzzy Inference System where linguistic variables, membership functions and rules were identified that use information from the three contextual dimensions (user, task, and environment). A high-level description of the classifier is presented in Table 3.

3.4. Initial design of the interruption classifier

This section presents the initial design of the Interruption Classifier using an Adaptive Neuro-Fuzzy Inference System. Workload levels for the game task are based on the number of jumping characters currently on the screen that need to be managed by the participant and the QueueSize represents how many interruption tasks are waiting to be served. Fig. 5 presents the surface view of initial model. Table 4 shows the initial fuzzy rules in the model.

3.5. Participants

Twenty-eight volunteers were involved as participants in this experiment (3 involved in the pilot study and 25 in the full experiment).[5] In the full experiment, there were 21 males and 4 females. Participants had a median age of 21 (mean 26.5, min. 19, max. 49). All participants were sampled from the general population of Sheridan College. Participants were recruited by a set of posters posted at various locations throughout the campus. The recruitment message did not disclose the purpose of the experiment, but described the task as fun and like a video game. The message indicated that each volunteer would receive compensation for his/her time and that volunteering would be a contribution to the advancement of science.

3.6. Analysis

Two types of analysis were performed on the collected data—-quantitative and qualitative analysis. We report only on the quantitative analysis as it is most closely tied to the design and evaluation of the classifier. Three types of quantitative analysis were performed: (1) determining the participant's performance using statistical tools; (2) evaluating and refining the classifier using standard machine learning evaluation tools; and (3) determining the accuracy of the classifier using statistical tools:

1. Computing confusion matrices: Compare the classifier's accuracy at predicting interruption points using the participant's timings as ideal. Confusion matrices and the associated measures (Accuracy, Precision and Sensitivity) are commonly used in the evaluation of machine learning algorithms, please see: (Elkan, 2011; Forman and Scholz, 2010; Hamilton, 2011; Kohavi and Provost, 1998; Lu et al., 2004). Statistics for each participant for each classifier model was collected on:

[5] This size of this experiment (in terms of the number of participants involved) is comparable to other similar studies in HCI; please see: Iqbal, S., & Bailey, B. (2006). Leveraging Characteristics of Task Structure to Predict the Cost of Interruption. Paper presented at the CHI 2006, Montreal, Quebec, Canada. (**n=12**). Lisetti, C. L., & Nasoz, F. (2004). Using Noninvasive Wearable Computers to Recognize Human Emotions from Physiological Signals. J. Appl. Sig. Process., 11, 1672-1687. (**n=14**). Scheirer, J., Fernandez, R., Klein, J., & Picard, R. W. (2002). Frustrating the user on purpose: a step toward building an affective computer. Interacting with Computers, 14, 93-118. (**n=24**).

True Positives (TP): (classifier correctly interrupted at the right time);
True Negatives (TN): (classifier correctly selected “Do not interrupt”);
False Negatives (FN): ("Didn't interrupt when it should have");
False Positives (FP): (“False Alarm”—classifier interrupted when it shouldn't have);

$$\text{Accuracy} = \frac{(True\ Positives + True\ Negatives)}{(True\ Positives + True\ Negatives + False\ Negatives + False\ Positives)};$$

$$\text{Precision} = \frac{True\ Positives}{(True\ Positives + False\ Positives)};$$

$$\text{Sensitivity} = \frac{True\ Positives}{(True\ Positives + False\ Negatives)}.$$

1. Calculating standard descriptive statistics on the confusion matrix data for the entire group including minimum, maximum, mean, median, standard deviation, accuracy, precision, and sensitivity.
2. Interruption timing analysis using a *bin* framework: Let the participant group, G, represent the group of all participants in the experiment, so $G = \{P_1,P_2,P_3,\ldots, P_n\}$,where n is the number of participants. For a given participant, P, a set of interruption timings is represented as: $\{int_1,int_2,int_3,\ldots, int_m|int_j \in \mathrm{N},\ 1 \le j \le m\}$. Each interruption timing, int_j, is measured in milliseconds since the beginning of an experimental trial for that participant. Thus, for the entire group of participants (i.e., G), a two-dimensional array is created: $timings_{P_i} = \{int_i^{(1)}, int_i^{(2)}, int_i^{(3)},\ldots,int_i^{(m)}\},\ where,\ \ 1 \le i \le n,$ and m represents the maximum number of interrupts in the experiment. A balance between high and low timing granularity is needed to facilitate a comparison between the Interruption Classifier timings and random assignment. A *bin* is used to represent a discrete section of time wherein one or more interruptions may occur. The *bin* analysis framework has been used in similar studies (Dobrian et al., 2011; Rind et al., 2011). Fig. 6 depicts a hypothetical scenario of interruption timings from random assignment, the Interruption Classifier, and the participant's interruption timings. A set of bins, $B = \{B_1,B_2,B_3,\ldots, B_k\},\ where\ k = |B|$ were constructed for this analysis. Note, k represents the mechanism that balances high and low granularity of interruption timing comparisons. Thus, for a given participant, P_i, $B_i = bins_{P_i} = \{bin_1^{(i)}, bin_2^{(i)}, bin_3^{(i)},\ldots,bin_k^{(i)}\},where\ 1 \le i \le n,$ and k is the maximum number of bins in the experiment. For our analysis, we used a bin size of 2 seconds, resulting in 135 bins over the 270 s trial period. The random assignment scheme (top section of Fig. 6) uses the total number of interruptions performed by the participant and randomly distributes them amongst the bins. Table 5 was used to record the random assignment timings, the classifier's timings, and the participant's interruption timings. If the classifier's predictions and the participant's timings fall into the same bin (i.e., both are *True*) or the participant did not serve an interruption and the classifier did not predict one (i.e., both are *False*), then this is classified as true positives and true negatives respectively; all other cases are classified as errors (an XNOR Boolean operation, denoted by ⊕—). Thus, the accuracy of the classifier's performance for a given P and trial is computed as:

Table 5
Template of interruption timings used for analysis for the classifier's performance compared to random and the participant's interruption timings.

Participant: P_i, Number of bins (k): 135, *bin* size: 2 s

	bin_1	bin_2	bin_3	…	…	…	…	bin_k
Random assignment	T	F	F	F	T	T	T	T
Interruption classifier	F	T	T	T	F	T	F	F
$P_i's$ interruption timings	F	T	T	T	F	T	F	F

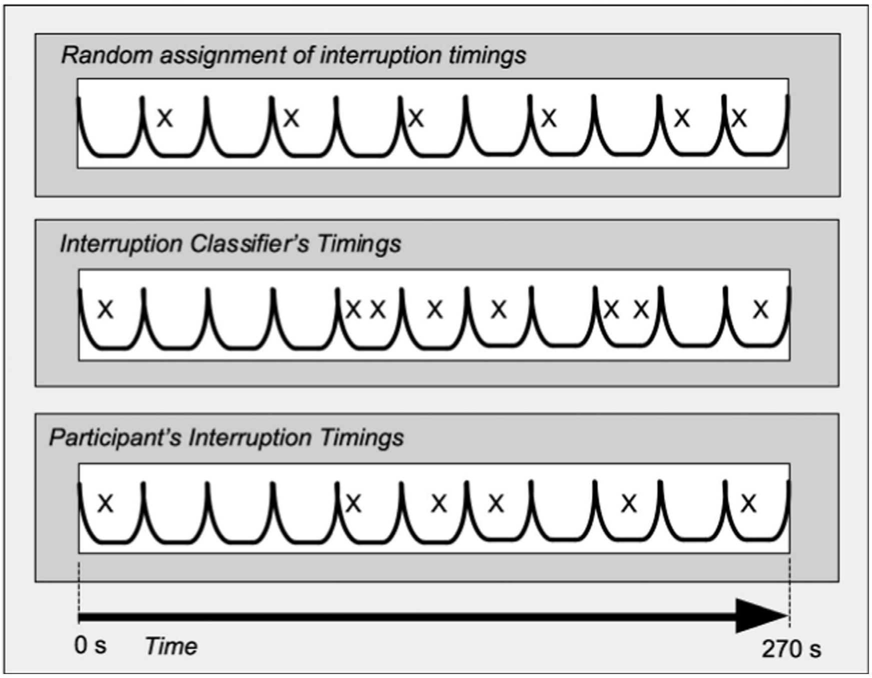

Random assignment of interruption timings
Interruption Classifier's Timings
Participant's Interruption Timings
0 s
Time
270 s

Fig. 6. Bin analysis framework to determine the performance of the interruption classifier. Bins are used to represent discrete equal-valued sections of time for the duration of an experimental trial. Interrupts are represented as X's in the bins.

$$\frac{\sum_{i=1}^{k} |\overline{Classifier_{timing\_bin_i} \oplus Participant_{timing\_bin_i}}|}{k} \qquad (1)$$

4. Findings (analysis and evaluation)

This section presents the findings (analysis and evaluation) of the research conducted. The topics include full experiment findings and the evaluation of the Interruption Classifier.

4.1. Summary of main findings

This section presents a summary of the main findings from design to refinement of the classifier. The following model variations were created and performed quite well. The main findings were:

- Different training data sets had the largest impact on the models created. Several participant data sets produced models that met and exceeded the performance criteria (6 out of 25). These models performed well across the entire participant group (generalizable to the sample population).
- Different time slices did not have a significant impact on the performance of the models created. All the constructed time slices (i.e., 1, 5, 10, 20, 25, 30, 50, and 100 ms) produced models that met the performance criteria. This may largely be due to the fact that these time slices are within the human physical response time (Hancock and Meshkati, 1988; Warm et al., 1996).
- The impact of historic event knowledge is significant. Knowing the past, especially when it is 100 ms or less in the past has a large impact in reasoning and deciding whether to interact with the user at a given point in time. All models created that used historic knowledge performed extremely well: the number of cases correctly classified was very high (> 99%); the TPs and TNs were very high (> 96% and 99% respectively); the FNs and FPs were very low (< 4% and < 0.7% respectively); and the models were accurate (> 98%) with little variation in the measures ($1.305 \le \sigma \le 5.793$)).
- Incorporating characteristics from the user model had a positive impact on the models created. The models that exceeded the performance criteria were: Initial model augmented with *tolerance to being interrupted*; Initial model augmented with *frustration level*; and Initial model augmented with *distractibility level*.

4.2. Full experiment findings

Data from the participants' trials were used in the analysis to determine the participant's performance. Observations recorded during this experiment were used to explore the timings of user-initiated interruption points. Furthermore, during this experiment, observations were recorded on the details of the context in which the people chose to serve interruptions.

Table 6
Descriptive statistics of number of jumpers saved (n = 25).

	Mean	Std. dev.	Min.	Max.	Median	Skewness	Kurtosis
Baseline – primary task only	37.58	3.414	29	43	37.5	−0.90	0.32
Baseline – interruption task only	0	0	0	0	0	0	0
Negotiated	35.18	3.619	28	42	35	−0.29	−0.49

Table 7
One-way ANOVA: between treatment condition effects for baseline-primary only and negotiated for number of jumpers saved.

Summary						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Primary task only Jsaved	25	939.5	37.58	12.139		
Negotiated Jsaved	25	879.5	35.18	13.643		
ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*p-value*	*F crit*
Between groups	72	1	72	5.585	0.022	4.043
Within groups	618.78	48	12.891			
Total	690.78	49				

Table 8
Descriptive statistics of matches done right of those attempted.

	Mean	Std. dev.	Min.	Max.	Median	Skewness	Kurtosis
Baseline – primary task only	0	0	0	0	0	0	0
Baseline – interruption task only	0.934	0.133	0.490	1	0.969	−0.96	1.49
Negotiated	0.914	0.129	0.490	0.99	0.959	−0.9	1.6

Table 9
One-way ANOVA: between treatment condition effects for baseline-game only and negotiated for number of matches correctly completed.

Summary						
Groups	*Count*	*Sum*	*Average*	*Variance*		
Interruption task-MRightOfDone	25	23.346	0.934	0.018		
Negotiated-MRightOfDone	25	22.855	0.914	0.017		
ANOVA						
Source of variation	*SS*	*df*	*MS*	*F*	*p-value*	*F crit*
Between groups	0.005	1	0.005	0.271	0.605	4.043
Within groups	0.856	48	0.018			
Total	0.861	49				

4.2.1. Number of jumpers saved on the game task

Table 6 presents the descriptive statistics.

A one-way ANOVA was performed to determine if there was

Table 10
Criteria for an acceptable model based on performance of the interruption classifier.

Criteria Attribute	Requirement
% Cases correctly classified	> 50%
% of interrupt now (True positive) cases	> 50%
% of do not interrupt (True negative) cases	> 50%
% of false negatives (Misses: didn't interrupt when it should have) cases	< 50%
% of false positives (False alarms: interrupted when it shouldn't have) cases	< 50%
Accuracy	> 50%
Precision	> 50%
Sensitivity	> 50%

Table 11
Confusion matrix results with supporting statistical measures and summative standard descriptive statistics for Participant #2 training data set for the initial model.

	Cases correctly classified (%)	True positive (%)	True negative (%)	False negatives (%)	False positives (%)	Accuracy (%)	Precision (%)	Sensitivity (%)
Min	39.843	79.134	24.676	1.643	0.378	80.086	94.116	79.134
Max	90.233	98.357	76.131	20.866	4.947	98.657	99.617	98.357
Mean	72.988	93.159	57.335	6.841	1.414	94.598	98.467	93.159
Median	**76.893**	**94.041**	**59.914**	**5.959**	**1.129**	**95.369**	**98.825**	**94.041**
Std dev	12.880	4.136	13.858	4.136	1.007	3.891	1.182	4.136

Table 12
Confusion matrix results with supporting statistical measures and summative standard descriptive statistics for Participant #21 training data set for the initial model.

	Cases correctly classified (%)	True positive (%)	True negative (%)	False negatives (%)	False positives (%)	Accuracy (%)	Precision (%)	Sensitivity (%)
Min	45.457	70.866	31.874	9.151	1.667	74.030	91.118	70.866
Max	89.940	90.849	79.045	29.134	6.908	93.443	98.092	90.849
Mean	74.837	83.400	60.785	16.600	3.344	87.684	96.093	83.400
Median	**78.577**	**83.938**	**61.508**	**16.062**	**3.329**	**88.423**	**96.202**	**83.938**
Std dev	11.581	4.903	12.652	4.903	1.144	4.232	1.510	4.903

difference in performance in the treatment conditions for the number of jumpers saved by treatment condition (Baseline–Primary Task only and Negotiated). Table 7 presents the results. There was a statistically significant difference between the groups Baseline—Primary Only and Negotiated performance at the 0.05 level, $F(1, 48) = 5.585$, $p = 0.022$. This means the additional activity to serve interruptions reduced the participants' performance in the primary task (i.e., being able to save jumping characters).

4.2.2. *Number of matches done right of those attempted*

Table 8 presents the descriptive statistics (mean, standard deviation, min., max., median, skewness and kurtosis).

A one-way ANOVA was performed to determine if there was difference in performance in the treatment conditions for the matches done correctly of those attempted by treatment condition (Baseline – Interruption task only and Negotiated). Table 9 presents the results. There was no statistically significant difference at the 0.05 level, $F(1, 48) = 0.271$, $p = 0.605$. This means the participants' performance level in the interruption task is independent of whether it is performed as the sole task or whether it is as an interruption task within performing the primary task.

4.3. *Interruption classifier evaluation*

This section presents the evaluation of the classifier. The criteria that was used for classifying a model as acceptable based on its performance is presented in Table 10. These criteria have been used in similar studies in Human Computer Interaction (Please see: (Finn and Limerick, 2003; Horvitz et al., 2004; Lisetti and Nasoz, 2004; Picard, 2003; Pu et al., 2006; Scheirer et al., 2002)).

Table 14
Descriptive statistics of interruption classifier performance (accuracy) using bin analysis.

	Mean	Std. dev.	Min.	Max.	Median	Skewness	Kurtosis
Random assignment	0.506	0.046	0.422	0.585	0.504	0.208	−0.463
Interruption classifier	0.962	0.033	0.881	1.000	0.970	−1.229	1.038

4.3.1. *Statistical analysis to test the classifier*

This section presents the statistical analysis that was performed to test the effectiveness of the classifier. Numerous confusion matrix computations were performed including supporting statistical measures and summative standard descriptive statistics. The confusion matrices show the range of values (min, max, mean, median and standard deviation) across all 25 participants. Three of these computational results are shown in Tables 11–13 based on models created from training data sets from Participant #2, #21, and #22 respectively (the others are omitted because they are quite similar).

After reviewing all the confusion matrix computations and supporting statistical results, it was discovered that the following models exceeded the performance criteria (all of the criterions shown in Table 10): 6 models based on Participant #2, #4, #6, #17, #21, and #22 training data sets; 3 models based on incorporating user characteristics: Initial model augmented with *tolerance to being interrupted*; Initial model augmented with *frustration level*; Initial model augmented with *distractibility level*; and several models based on the initial model

Table 13
Confusion matrix results with supporting statistical measures and summative standard descriptive statistics for the Participant #22 training data set for the initial model.

	Cases correctly classified (%)	True positive (%)	True negative (%)	False negatives (%)	False positives (%)	Accuracy (%)	Precision (%)	Sensitivity (%)
Min	46.419	81.037	27.957	3.671	0.799	86.004	95.911	81.037
Max	90.013	96.329	79.045	18.963	3.499	96.816	99.169	96.329
Mean	75.081	89.024	60.034	10.976	2.128	92.035	97.634	89.024
Median	**79.031**	**88.781**	**61.391**	**11.219**	**1.992**	**92.134**	**97.760**	**88.781**
Std dev	11.482	4.477	13.382	4.477	0.771	2.841	0.929	4.477

Table 15
Confusion matrix results with supporting statistical measures and summative standard descriptive statistics for the Participant #2 training data set for the initial model with PreviousTimeStep.

	Cases correctly classified (%)	True positive (%)	True negative (%)	False negatives (%)	False positives (%)	Accuracy (%)	Precision (%)	Sensitivity (%)
Min	94.994	73.879	93.807	0.757	0.149	83.843	92.265	73.879
Max	99.876	99.243	99.851	26.121	6.193	99.547	99.850	99.243
Mean	99.011	94.271	98.797	5.729	1.203	96.534	98.667	94.271
Median	**99.407**	**96.738**	**99.274**	**3.262**	**0.726**	**98.018**	**99.255**	**96.738**
Std dev	1.061	5.793	1.305	5.793	1.305	3.539	1.598	5.793

with historic event knowledge.

4.3.2. Determining the classifier's performance using the bin analysis framework

This section presents the analysis that was performed to assess the classifier's performance based on the bin analysis framework. Table 14 presents the summary of computed accuracy (Eq. (1)) of the classifier involving all 25 participants using descriptive statistics. The classifier performed at 96% accuracy (mean), with a variation from 88% in worst case to 100% in the best case.

4.4. Summary of most successful and least successful models

4.4.1. Most successful models

Beyond the successful models presented earlier, models that incorporated historic information performed the best across the entire population. Table 15 presents the results of a model based on Participant #2 training data set with PreviousTimeStep. Twelve other models that used historic information performed similarly with very high accuracy.

4.4.2. Least successful models

During this multi-year research project, hundreds of models were created using different training data sets and combinations of properties from the user, task and environment dimensions, and classifier parameter settings. Most of these models performed unsatisfactorily. One of the models that performed poorly was based on Participant #8′s training data. This participant saved the fewest jumpers: 28/48 (population mean = 35.2), completed fewer interruption tasks: 24.5/50 (population mean = 45.5), and took longer than other participants to perform the interruption tasks: mean = 13.3 s (population mean = 3.6 s). Furthermore, it was also discovered that this participant did not execute a consistent strategy with respect to choosing appropriate times to serve interruptions. Table 16 shows the confusion matrix results of a model based on this participant.

Another poorly performing model was based on Participant 25's training data. This person reported “very low tolerance to interruptions,” “very high level of frustration,” “very high level of distractibility,” and “very difficult to regain focus [after an interruption].” Table 17 shows the results of a model based on this participant.

5. Discussion

This section presents a summary of the significant factors in designing good models; predictive factors of interruptible moments; implications of the empirical results for theories of interruption; and implications of the empirical results for deploying classifiers for detecting interruptible moments in practice.

5.1. Significant factors in the design of good models

- *Training datasets:* Good training data sets had the largest impact on creating good models. Several participant data sets (6 out of 25) produced models that exceeded the performance criteria (see Table 10). These models performed well across the entire participant group (generalizable to the sample population). Unfortunately, there is no way to determine from the outset if a training data set will provide the foundation for a good model. The process of presenting it to the machine learning tool for training, testing the model created and then analyzing the results is required to determine if it good or not.
- *Historic information*: The impact of historic event knowledge is significant. Knowing the past, especially when it is 100 ms or less in the past has a large impact in reasoning and deciding whether to interact with the user at a given point in time. All models created that used historic knowledge performed extremely well: the number of cases correctly classified was very high ($> 99\%$); the TPs and TNs were very high ($> 96\%$ and 99% respectively); the FNs and FPs were very low ($< 4\%$ and $< 0.7\%$ respectively); and the models were accurate ($> 98\%$) with little variation in the measures ($1.2 \le \sigma \le 6.0$)).
- *User model*: Incorporating characteristics from the user model had a positive impact on the models created. The models that exceeded the performance criteria were: Initial model augmented with *tolerance to being interrupted*; Initial model augmented with *frustration level*; and Initial model augmented with *distractibility level*.
- *Machine learning*: An appropriate machine learning tool is necessary to create effective, well performing models. The classifier is based on an Adaptive Neuro-Fuzzy Inference Systems machine learning tool. The ANFIS is an advanced hybrid soft computing tool that uses fuzzy logic and artificial neural networks (Jang et al., 1997). The classifier created during this research is the first of its kind based on

Table 16
Confusion matrix results with supporting statistical measures and summative standard descriptive statistics for the Participant #8's training data set for the initial model.

	Cases correctly classified (%)	True positive (%)	True negative (%)	False negatives (%)	False positives (%)	Accuracy (%)	Precision (%)	Sensitivity (%)
Min	78.016	0.000	76.314	54.173	10.885	40.154	0.000	0.000
Max	90.189	45.827	90.138	100.000	27.738	64.849	78.107	45.827
Mean	83.880	7.008	82.630	92.992	18.850	44.551	22.399	7.008
Median	**83.862**	**4.179**	**83.561**	**95.821**	**19.244**	**43.235**	**17.606**	**4.179**
Std dev	2.984	8.935	3.405	8.935	4.177	4.780	17.274	8.935

Table 17
Confusion matrix results with supporting statistical measures and summative standard descriptive statistics for the Participant #25's training data set for the initial model.

	Cases correctly classified (%)	True positive (%)	True negative (%)	False negatives (%)	False positives (%)	Accuracy (%)	Precision (%)	Sensitivity (%)
Min	77.218	0.000	73.895	79.212	10.459	38.873	0.000	0.000
Max	90.064	20.788	89.845	100.000	28.279	49.619	52.536	20.788
Mean	83.182	4.602	82.396	95.398	19.423	43.134	17.604	4.602
Median	**83.605**	**2.975**	**83.261**	**97.025**	**19.244**	**42.381**	**13.856**	**2.975**
Std dev	3.055	4.475	3.432	4.475	4.195	2.673	13.764	4.475

ANFIS for HCI purposes. Please see Appendix F: Functional Description of the Interruption Classifier for more details about the machine learning algorithm used in this research.

- *Generalizable*: The classifier is generalizable to a degree. We discovered that user modeling information such as personality traits are generalizable across all models in all the experiments conducted. However, other aspects of the modeling process (e.g., task details), need to be explicitly represented in a model for it to be successful and are therefore not directly generalizable.

5.2. Predictive factors of interruptible moments

In this research, both quantitative and qualitative data were collected and analyzed. Since the paper primarily focused on the quantitative aspects, this section presents some of the insights from the qualitative data as to why users preferred certain points for interruption. One of the goals in the qualitative area of research was to shed some light onto the motivation and context to answer the question “Why now?” —that is, why did participants choose those specific points to serve interruptions and what was the context surrounding that decision. The following selected anecdotes are from the closing questionnaires from Participant #2 and Participant #17 whose data sets produced good models (Appendix D presents all of the participant's strategies with respect to interruption timings):

``I tried to position the paramedics in a logical place and tried to switch when people were bouncing up. I picked people that were closer to the ambulance when deciding who would be saved/not saved. For the matching task, I just focused on the word, and let my peripheral eyesight decide [colour vs. shape].” (Participant #2).

``The matching [interruption] task by itself and the game [primary] task by itself were very simple. Together though it was more difficult. I would position my stretcher so the jumpers would be secure while I switched tasks.” (Participant #17).

Researcher observation and analysis showed that both participants followed the same advanced strategies when compared to other participants:

- In the primary task, the character on the third bounce has increased value in that the participant has invested effort and time in saving the character thus far. So, when faced between saving a new jumping character from the building versus saving the character on the third and final bounce into the ambulance, the participant would position the stretcher at the last bounce location to fulfil this goal.
- The participants would keep track of the number of interruption tasks waiting in the queue by counting the number of flashes on the screen. They would wait until the primary task was sufficiently stable (i.e., few jumpers to manage or several jumpers to manage but they were all nicely in the air) and then perform the switch and complete all the required interruption tasks.

These strategies enabled the participants to perform well on both the primary and interruption tasks: Participant #2 saved 90% of the jumpers, and completed 93% of the interruption tasks correctly; Participant #17 saved 85% of the jumpers, and 99% interruption tasks were completed successfully. (Their performance was statistically significantly higher than other participants – across the entire study the mean number of jumpers saved was: 73%, and the mean number of interruption tasks completed successfully was: 91%). In regards to predictive factors of the interruptible moment, these strategies were consistently and accurately performed and were implicitly recorded in the participants' data sets which served well in training the classifier.

Regarding the participants' user models, both Participant #2 and #17 reported on their opening questionnaires a strong capacity to perform well on the upcoming experiment tasks (please see Appendix B: Opening Questionnaire). Table 18 presents these participant's personal characteristics and the means across all participants.

On the closing questionnaires, both Participant #2 and #17 reported: *Performed well* (100%) and *Picked the right time for serving the interruption* (100%) (please see Appendices C: Closing Questionnaire). Across the entire study the means for *Performed well* was: 78%, and *Picked the right time for serving the interruption* was: 84%.

In developing the models for the classifier, numerous combinations of characteristics were used (*ToleranceToInterruptions, Age, VideoGameFamililiarity, Frustration, Distractibility, MultitaskingAbility, GeneralComputerFamiliarity*, and *ThinkingStyle*). Through experimentation many of these models were discarded, however, the models that performed well were the Initial model augmented with *tolerance to being interrupted*; Initial model augmented with *frustration level*; and Initial model augmented with *distractibility level*. These personal characteristics were found to be significant in designing models that were predictive of good interruption points.

This section presented some of the insights from the qualitative data as to why users preferred certain points for interruption. It also showed that the predictive factors of interruptible moments for free-form tasks is very dependent on well-defined data sets that accurately represent the specific task details, strategies and nuances, coupled with good user models. These data sets are essential for training in the machine learning process to create good classifiers.

5.3. Implications of the empirical results for theories of interruption

Based on the empirical results in this research there are several implications to theories of interruption. Gievska's theory focused exclusively on task-based contextual information (Gievska and Sibert, 2005). These researchers stated that additional work needs to be done in this area by including information from the other dimensions in their taxonomy: “Subjective preferences were not considered as a factor in the current implementation of the interruption mediator. However, they should not be neglected when designing user interfaces that give equal priority to user's satisfaction and comfort as to other performance measures” (Gievska et al., 2005, pg 176). One specific implication of the empirical results is that Gievska's theory of interruptions could be updated to emphasize the characteristics that were found to be significant in this research, namely: *tolerance to interruptions; frustration*

Table 18
Personal characteristics for Participant #2 and #17 and the means across all participants.

	General Computer familiarity (100 % = very familiar)	Familiarity with video games (100 % = very familiar)	Tolerance to interruptions (100% = don't mind interruptions)	Frustration level (0 % = not easily frustrated)	Distractibility level (0 % = not easily distracted)	Thinking style (0% = thinker, 100% = hands-on)	Multi-tasking ability (100% = very good at multi-tasking	Regaining focus ability (0% = easily regains focus after interruption)
Participant #2 and #17	100%	100%	100%	0%	0%	60%	100%	0%
Mean	90%	83%	68%	54%	62%	64%	58%	53%

level; and *distractibility level.*

The empirical results from this study also highlight the importance of well-defined rules that capture the salient features of the tasks. The models that performed the best in this study incorporated good task management strategies in their design. From this perspective, it reinforces the theory that task properties and details are fundamentally important in designing systems that determine good interruptible moments. This perspective is well represented in the literature on theories of interruption. Beyond these implications, there are many opportunities for addition research to be conducted in the spirit of refining these theories.

5.4. Implications for practice

This research showed that there is significant potential to create classifiers using machine learning to determine appropriate times to interrupt the user. In regards to deploying machine learning classifiers in practice, there are a number of opportunities through which we may witness a significant uptake on classifiers that improve our quality of experiences with computers and computing devices (mobile devices, wearable devices, etc.) that determine good interruption points.

5.4.1. Considerations for a robust multi-user classifier

The next step in developing this classifier for use in practice would be to create a system of classifiers in which each classifier would be responsible for learning and personalizing the interruption strategy for a specific user for a specific set of tasks the user routinely performs. In this way, the classifier would be able to tune its behaviour to the specific user it is assisting.

The current method for creating satisfactory models requires substantial manual effort on the part of the researcher to train a classifier, analyze the results and then select the best performing models that meet the requirements. Further work is needed to refine this process by designing an automated solution (*model_evaluator*) that would receive the interactive datasets, generate and analyze a variety of models, and then rank and select the best model from this list.

In terms of implementation, a cloud-based system would be a suitable architecture to receive interruption and associated contextual information to support each user. This research focused on a primary and interruption task, however, in a practical implementation, rather than collecting datasets from a variety of people performing the same tasks, it would be more effective for many unique classifiers to be created primarily based on datasets from individuals. In this way, a variety of classifiers would learn based on personalized models and thus be better attuned to each user. Each classifier would learn its user's personal interruption characterization profile of when s/he prefers to be interrupted and when to be left undisturbed. Once the cloud service has a sufficient amount of data from a user, the machine learning algorithm would be trained and models created. The overarching module (*model_evaluator*) would evaluate and tune the machine learning parameters and attempt to improve the models for that specific user. The best performing model that meets the requirements (see Table 10) would be used for determining the timing of interruptions for that user. If none are acceptable, the cloud-based system continues to collect additional interaction data. This cyclic process continues over time to refine the models generated and to persistently select the best model representative for that specific user. In this way, this system would provide multiple dynamically adjusting classifiers for each user for tasks that s/he performs on a regular basis.

5.4.2. Users learning to interact with the classifier

One of the key benefits of the classifier created in this work is that the user is not required to be involved in the classifier's learning process whatsoever. The interruption points are discovered by the classifier from the interaction datasets (e.g., the task switches from the primary task to the interruption task among other contextual information). This desirable characteristic is one of the main contributions of this work and essential for future versions of the classifier. We believe this approach is an appropriate direction of research due to two main powerful trends: 1) Big Data, and advances in machine learning, and 2) Computer Vision and facial feature detection.

With the advances in Big Data and machine learning there are many algorithms that can extract features from data that could be used to drive or enhance the operation of the classifier. It is foreseeable the associations that are not obvious to us may in fact be readily uncovered by an ML algorithm that can identify if a user is becoming frustrated, confused, or is concentrating intensely, etc. This input in combination with task understanding could aid the Interruption Classifier with an increased ability to interrupt the user more accurately and in more widespread scenarios.

Additionally, with advances in computer vision and facial feature analysis, a future version of the Interruption Classifier could accept input such as eye-tracking information (e.g., saccades,[6] perclos, fixations, gazepoints, percentage of time the user focused on tasks and subtasks, pupil diameter, and blink rates), and facial features (e.g., head positions and rotations; eyelid aperture; lip and eyebrow movements with appropriate inferences (e.g., smiling, frowning, concentrating, etc.). These feature rich details provide significant insight into the cognitive state of a person and may be key indicators to assist in picking appropriate times to interrupt the user (Davison et al., 2018; Reisenzein et al., 2017). These inputs, in combination with the other data collected in this study, would provide a robust version of our classifier a more holistic view of the user and the tasks that s/he is performing, ultimately enabling the classifier to better learn and reason about ideal interruption points.

5.4.3. Content and relevance of the tasks

The content and nature of the tasks the user is performing needs to be precisely represented (as in this study with the various task rules) for the Interruption Classifier to make good decisions about ideal interruption points. In this research, the primary and interruption tasks and user contexts were well represented in the machine learning algorithm. The primary task had a substantial number of rules that collectively captured ideal situations for when an interruption could occur. However, by comparison, there were significantly fewer rules embodying the task characterization for the interruption task.

In practice where users are regularly multi-tasking, it may be difficult to determine what the primary / interrupting task is. In these scenarios, when the distinction between the primary and interruption tasks may not be obvious, the same degree of rigor is required to represent the tasks the user routinely performs. In this way the classifier would have models of all relevant tasks in play and can more effectively reason about making interruption decisions. Essentially, the rule base for both the primary and interruption tasks would be equally well-defined so that interruptions could occur from within either task. This approach could be extended to multiple tasks the user may perform.

5.4.4. User productivity and contentment

A natural question to ask regarding this work is: “Would users become more productive and/or more content over time?” Theoretically, the classifier is generating interruptions at points when the user would have selected them. As mentioned earlier, this classifier is not concerned about optimizing the user's productivity, but rather choosing times that most appropriately coincide with the user's preference while considering contextual factors such as task and user characteristics. While our classifier may not explicitly try to optimize the user's productivity, it is a plausible outcome because the classifier is in tune with that person's personal interruption profile. As a result, the user's overall satisfaction with respect to interruptions could be said to be optimized. If the user is more content, then the user's productivity may well increase too (Li and Fuller, 2017; Lucassen et al., 2017).

5.5. Limitations and future work

This research showed that there is significant potential to create classifiers using machine learning to determine appropriate times to interrupt the user. This section presents some of the limitations and goals for future work.

Limitations

As with any research there are limitations. Some of the main limitations of this work are:

1. When good training datasets are used, the classifier's accuracy is increased significantly. A limitation of the classifier is that finding good training datasets in complex real-world problem domains may be difficult to acquire. We believe we can address this by implementing the cloud-based system proposed earlier where each user of the system seamlessly provides interaction data and task-based contextual information to enable classifiers to learn, improve and adapt over time.
2. The number of participants involved in this study was sufficient for this research, however, it would be interesting to find out how the classifier performs with a large population, for instance, what is its’ performance when 1000 participants are involved? Also, it would be interesting to find out how the models work with older participants (e.g., seniors) or with user models vastly different from those used in this study.

Future work

5.5.1. Mobile device interruption management services

The field of mobile computing is growing at a staggering rate and the problem of interruptions is even more poignant in this context (Pielot et al., 2014; David et al., 2015; Sarker et al., 2017). Not until quite recently did smartphones provide support for developers to customize user notifications and interruptions (Pielot et al., 2017). The classifier could be extended to draw from the user model and rich set of sensory components on these devices (accelerometer, GPS location, microphone, calendar, etc.). These additional input sources may enhance the performance of the classifier to increase user satisfaction and productivity. Building on the findings of this research, future studies should explore how the classifier could be adapted and improved to serve in the context of mobile devices.

5.5.2. Intelligent personal assistant

In alignment with the current trends of cognitive computing and personal assistants (e.g., Siri, Cortana, etc.), the classifier could be extended to observe what the user is doing in a broader more comprehensive perspective. For example, suppose the user is working on a paper that involves many activities and the classifier observes that the user has attempted to perform the same activity many times within a ½ h period. The classifier reasons about the ideal interruption point and decides to issue an interrupt message, such as, “Excuse me, Sue, it appears that you have you been trying ⟨*name of task*⟩ repeatedly now for 30 min – the following ⟨*suggested action*⟩ will accomplish the same goal more quickly.” Since the machine learning tool used in this research (i.e., ANFIS) contains details about its reasoning process, an appropriate detailed message can be provided. The classifier is well-designed to support future work regarding presenting specific interruption

[6] A saccade is an abrupt quick but short movement in both eyes. An Eye Tracker records major saccades (noticeable to the naked eye) and minor saccades (where special instrumentation is needed to detect them). Perclos is the percentage of eyelid closure over the pupil.

messages that are personally relevant and of substantial utility to the user's current context.

5.5.3. Enhanced personal modeling

This research showed that some user characteristics (e.g., *ToleranceToInterruptions*) are significant in the creation of good classifiers. Future research that aims to deploy machine learning classifiers for detecting interruptible moments in practice should focus on collecting additional user characteristics such as, biometrics from wearable devices (e.g., $SP0_2$ pulsioximeter, spirometer, blood pressure, EMG, ECG, temperature, etc.), and enriched personality trait attributes (e.g., Myers–Briggs personality test). These additional inputs in combination with task and environment details may result in classifiers that offer improved accuracy, performance and relevance to users' interruption experiences.

6. Conclusion

Determining when to interrupt a user at appropriate times as s/he performs computer-based tasks is an ongoing problem to which we have provided a solution. We created a classifier that contributes to the field in the following ways:

1. The classifier incorporates a user model in its' reasoning process. Most interruption systems focus on task-based contextual information only. Our classifier includes user and task contextual information.
2. The classifier performs better than random, and, in the best models constructed, performs at an accuracy of 98% with historic event knowledge and 95.4% without historic knowledge (Tables 11–13).
3. User modelling integration with machine learning algorithms was explored and appears to be very promising.
4. The classifier was implemented using an advanced machine learning technology (ANFIS)—which is a novel contribution. No other interruption system uses an ANFIS.
5. This research sheds light on reasoning about ideal interruption points for free-form tasks. Currently, this is largely an unsolved problem.

This research also assessed the participant's performance at the tasks and evaluated the classifier's performance. It was demonstrated that many models performed extremely well. The classifier was designed with a framework so that it could be generalized to other tasks and problem domains.

In the spirit of furthering science and this work, the MATLAB source code for the classifier, models, and data sets will be openly available on the author's and/or journal's website. We hope this will encourage other researchers to extend and explore our work and to test and compare our classifier with other interruption systems.

Appendices

This section presents supporting documents used in conducting this research. The appendices are:

- Appendix A: Experiment protocol
- Appendix B: Opening questionnaire – participant
- Appendix C: Closing questionnaire – participant
- Appendix D: Closing questionnaire – results: strategies and general comments
- Appendix E: ANFIS fuzzy rules
- Appendix F: Functional description of the interruption classifier
- Appendix G: Raw numbers for the cases "interrupt" relative to the total number of cases

Appendix A: experiment protocol

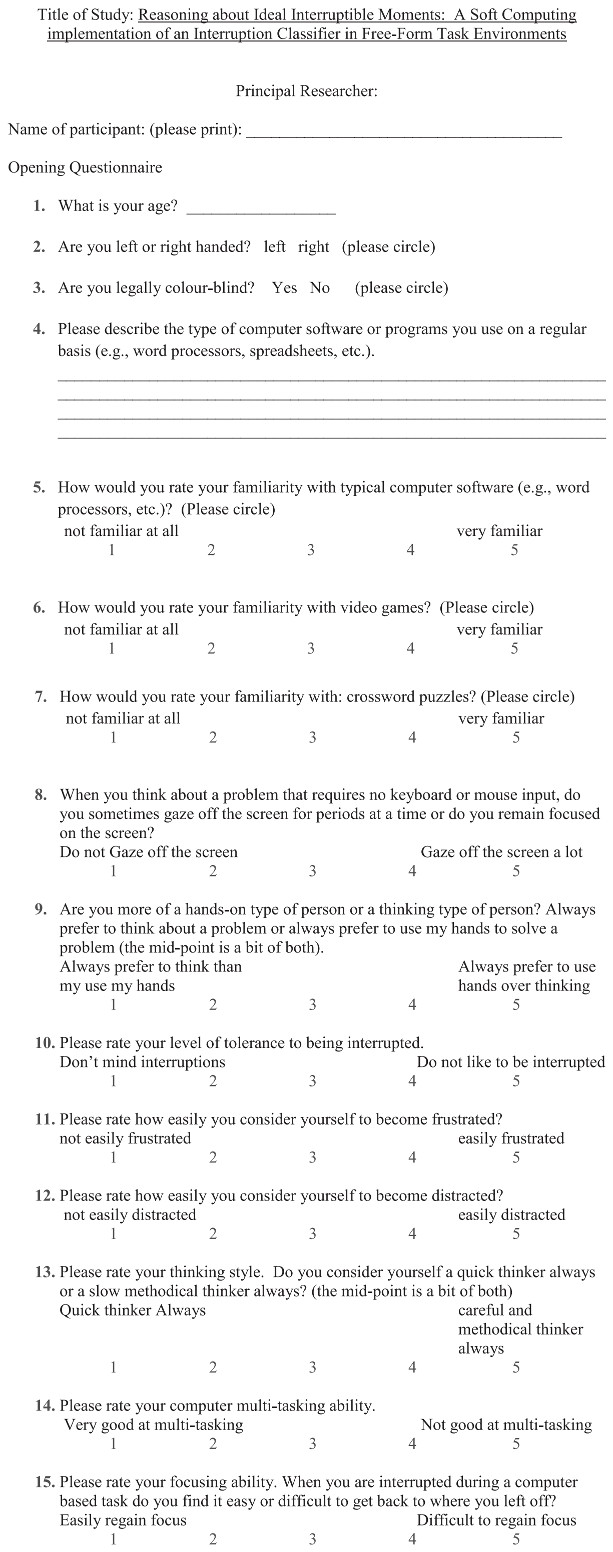
This appendix presents the experiment protocol that was used by the researcher to ensure the experiment was being conducted in a consistent way for all participants.

Experiment protocol

1. Greeting and introduction
2. Verify the participant has the minimal requirements to participate in this experiment: (a) Have you have normal color vision?" (b) "can you read English?" (c) "can you press keys on a computer keyboard with one hand?" and (d) "are you 18 years old or older?"
3. Acquire participant's signature on a consent form that explains their rights.
4. Administer participant opening questionnaire.
5. Ask participant to read the instruction sheet.
6. Ask participant to sit in a comfortable chair in front of a computer with the experimental tasks ready.
7. Set the `-s114 -g3` in `Run_Experiment.java`
8. Record this information on Opening Questionnaire.
9. Administer the 12 trials of the computer-based dualtask: The experimenter will sit near the back of the room so as not to interfere, but able to answer questions if necessary. **Encourage participant to ask questions during the practice sessions.**
10. Administer the Closing Questionnaire.
11. Debriefing—Ask participant to clarify their strategy used and explain why. Give participants their compensation.

Appendix B: opening questionnaire – participant

This appendix presents the opening questionnaire that was used to collect information from participants in this research.



Title of Study: Reasoning about Ideal Interruptible Moments: A Soft Computing implementation of an Interruption Classifier in Free-Form Task Environments

Principal Researcher:

Name of participant: (please print): ______________________________________

Opening Questionnaire

1. What is your age? __________________

2. Are you left or right handed? left right (please circle)

3. Are you legally colour-blind? Yes No (please circle)

4. Please describe the type of computer software or programs you use on a regular basis (e.g., word processors, spreadsheets, etc.).
 ____
 ____
 ____
 ____

5. How would you rate your familiarity with typical computer software (e.g., word processors, etc.)? (Please circle)
 not familiar at all very familiar
 1 2 3 4 5

6. How would you rate your familiarity with video games? (Please circle)
 not familiar at all very familiar
 1 2 3 4 5

7. How would you rate your familiarity with: crossword puzzles? (Please circle)
 not familiar at all very familiar
 1 2 3 4 5

8. When you think about a problem that requires no keyboard or mouse input, do you sometimes gaze off the screen for periods at a time or do you remain focused on the screen?
 Do not Gaze off the screen Gaze off the screen a lot
 1 2 3 4 5

9. Are you more of a hands-on type of person or a thinking type of person? Always prefer to think about a problem or always prefer to use my hands to solve a problem (the mid-point is a bit of both).
 Always prefer to think than my use my hands Always prefer to use hands over thinking
 1 2 3 4 5

10. Please rate your level of tolerance to being interrupted.
 Don't mind interruptions Do not like to be interrupted
 1 2 3 4 5

11. Please rate how easily you consider yourself to become frustrated?
 not easily frustrated easily frustrated
 1 2 3 4 5

12. Please rate how easily you consider yourself to become distracted?
 not easily distracted easily distracted
 1 2 3 4 5

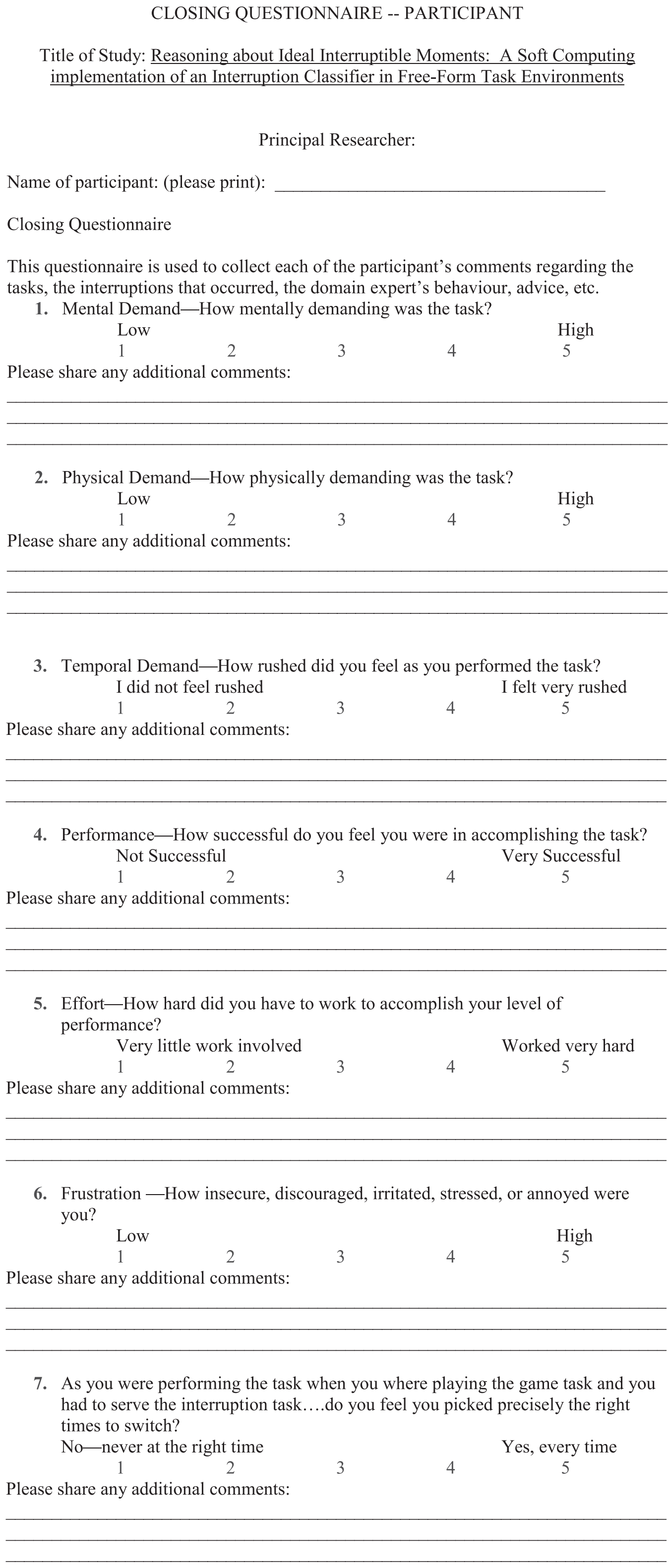
13. Please rate your thinking style. Do you consider yourself a quick thinker always or a slow methodical thinker always? (the mid-point is a bit of both)
 Quick thinker Always careful and methodical thinker always
 1 2 3 4 5

14. Please rate your computer multi-tasking ability.
 Very good at multi-tasking Not good at multi-tasking
 1 2 3 4 5

15. Please rate your focusing ability. When you are interrupted during a computer based task do you find it easy or difficult to get back to where you left off?
 Easily regain focus Difficult to regain focus
 1 2 3 4 5

Appendix C: closing questionnaire – participant

This appendix presents the closing questionnaire that was used to collect information from participants after completing the experiment.



CLOSING QUESTIONNAIRE -- PARTICIPANT

Title of Study: Reasoning about Ideal Interruptible Moments: A Soft Computing implementation of an Interruption Classifier in Free-Form Task Environments

Principal Researcher:

Name of participant: (please print): ______________________________

Closing Questionnaire

This questionnaire is used to collect each of the participant's comments regarding the tasks, the interruptions that occurred, the domain expert's behaviour, advice, etc.

1. Mental Demand—How mentally demanding was the task?

Low High
1 2 3 4 5

Please share any additional comments:

____
____
____

2. Physical Demand—How physically demanding was the task?

Low High
1 2 3 4 5

Please share any additional comments:

____
____
____

3. Temporal Demand—How rushed did you feel as you performed the task?

I did not feel rushed I felt very rushed
1 2 3 4 5

Please share any additional comments:

____
____
____

4. Performance—How successful do you feel you were in accomplishing the task?

Not Successful Very Successful
1 2 3 4 5

Please share any additional comments:

____
____
____

5. Effort—How hard did you have to work to accomplish your level of performance?

Very little work involved Worked very hard
1 2 3 4 5

Please share any additional comments:

____
____
____

6. Frustration —How insecure, discouraged, irritated, stressed, or annoyed were you?

Low High
1 2 3 4 5

Please share any additional comments:

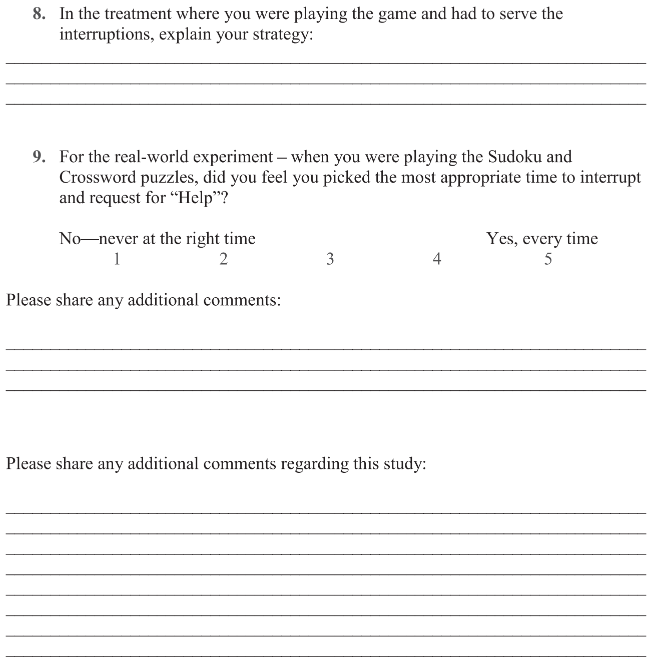
____
____
____

7. As you were performing the task when you where playing the game task and you had to serve the interruption task….do you feel you picked precisely the right times to switch?

No—never at the right time Yes, every time
1 2 3 4 5

Please share any additional comments:

____
____
____



8. In the treatment where you were playing the game and had to serve the interruptions, explain your strategy:

____
____
____

9. For the real-world experiment – when you were playing the Sudoku and Crossword puzzles, did you feel you picked the most appropriate time to interrupt and request for "Help"?

No—never at the right time				Yes, every time
1	2	3	4	5

Please share any additional comments:

____
____
____

Please share any additional comments regarding this study:

____
____
____
____
____
____
____
____

Appendix D: closing questionnaire – results: strategies and general comments

This appendix presents the strategies and general comments from selected participants (data extracted from closing questionnaires).

Participant #	Strategy and general comments
1	``I would bounce the character to make sure they are elevated in the air, and then serve the interruption."
2	``I tried to position the paramedics in a logical place and tried to switch when people were bouncing up. I picked people that were closer to the ambulance when deciding who would be saved/not saved. For the matching task, I just focused on the word, and let my peripheral eyesight decide." [colour vs. shape]
3	``If there were a lot of falling people, I would let the matching tasks queue up for a little bit."
4	``I usually served the interruption immediately after an interruption notification was issued (flash on the screen). This strategy, I feel, rarely affected me from saving people. I would put most of my efforts into saving people and deal with the interruptions as they came up."
6	``It was impossible to save all the people (jumping characters). Noted the number of matching interruptions that were queued (from the number of flashes on the screen) and switched from game to matching task so that most of the time I would save the jumpers and complete as many matching tasks as possible. Errors occurred when the estimated time to complete the match exceeded the time estimated, resulting in failure to save jumpers (only in tight situations)."
7	``My strategy was to save as many people by performing each interruption task right when notified to avoid a build-up of interruption tasks (matching tasks)."
8	``Wait for a period with a few jumpers or high jumpers, then go through the colour/shape task quickly. Sometimes, I let the number of matching tasks build up too much. At times it was easy, other times, I felt rushed and stressed. I experienced some eye strain, and staying focused was difficult."
9	``I tried to position the net/trampoline under oncoming people before activating the interruption task. Sometimes multiple matches at once would thwart this strategy."
10	``Anticipate where people would fall and place the stretcher (medics) before switching to the interruption task."
11	``I put the catching net in place in advance, then switched to the interruption task while they bounced up."
12	``The mental demand for the Game Task was 3/5, whereas the Matching Task was 5/5. Effort: It took most of my focus. I had to remind myself to blink. Serve interruptions when people are bouncing up. Leave stretcher in optimal place to catch (bounce) people before serving interruption task."
14	``Place the stretcher under the next fall so that I could do the interruption as they bounced up then go back to the game task."
15	``Try to line up stretcher with as many falling people and quickly do the matching task."

16	``I found the practice sessions very good – they were difficult, but near the end I had no trouble performing both the Game Task and Matching Task. I would position the stretcher in the spot that a Jumper would hit and then served the interruption task. Before I switched to serve an interruption, I made sure to remember the position of the falling people in my mind.”
17	``The Matching Task by itself and the Game Task by itself were very simple. Together though it was more difficult. I would position my stretcher so the jumpers would be secure while I switched tasks.”
19	``I tried to keep tabs on the characters, and only interrupted the game play when I knew they wouldn't be falling away from the stretcher. I usually let the matching tasks queue up for a bit, but if the queue was too big, then I would miss the timing for a jumper's rescue when I returned to the Game Task.”
20	`I tried to interrupt when most of the jumpers were ascending to give me time to complete the matching task but I tried not to queue more than three at a time. I am a person who does not like to lose and I was determined to get 100% even though I knew I couldn't save all the Jumpers.”
21	“Move to a spot where I could remain for a few seconds, then perform a matching task.”
22	“I found the tasks easy to perform. My strategy was to anticipate and predict when the characters will start to fall. I would do a matching task if the characters were bouncing up in the air.”
23	“Sometimes I tried to pick the interruption task quickly to get it out of the way, however, sometimes this is not the best strategy. I tried to be in the right position with the medics before I served an interruption. Got used to the control and feel of the Game (both Game Task and Matching Task) within 10 minutes. Frustrated that I couldn't save all the Jumpers.”
25	“Waited until the jumpers were in the air (bouncing up), then it was safe to serve the Matching Task.”

Appendix E: ANFIS fuzzy rules

This appendix presents the ANFIS Fuzzy Rules used during the development and refinement of the interruption classifier.

Table 1
ANFIS fuzzy rules—initial model.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 19
ANFIS fuzzy rules—initial model + JumperState rules.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 20
ANFIS fuzzy rules—initial model + JumperState rules (Comprehensive).

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 21
ANFIS fuzzy rules: initial model + ToleranceToInterruption.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 22
ANFIS fuzzy rules: initial model + ToleranceToInterruption + Age.

Rule #	Rule sescription
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 23
ANFIS fuzzy rules: initial model + Video_Game familiarity.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 24
ANFIS fuzzy rules: initial model + ToleranceToInterruptions + Video_Game familiarity.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 25
ANFIS fuzzy rules: initial model + ToleranceToInterruptions + frustration.

Rule #	Rule fescription
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 26
ANFIS fuzzy rules: initial model + frustration.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 27
ANFIS fuzzy rules: initial model + distractibility.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 28
ANFIS fuzzy rules: initial + tolerance to interruptions + distractibility.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 29
ANFIS fuzzy rules: initial model + ToleranceToInterruptions + MultiTaskingAbility.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 30
ANFIS fuzzy rules: initial model + PreviousTimeStep.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is least) then (interruptDecision is no1) (1)
2	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is most) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is great) then (interruptDecision is yes4) (1)
5	If (Workload is hard) and (QueueSize is least) then (interruptDecision is no2) (1)
6	If (Workload is hard) and (QueueSize is small) then (interruptDecision is no3) (1)
7	If (Workload is hardest) then (interruptDecision is no4) (1)

Table 31
ANFIS fuzzy rules—all rules model.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is small) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes1) (1)
2	If (Workload is easiest) and (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is small) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes4) (1)
5	If (Workload is easy) and (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes5) (1)
6	If (Workload is easiest) and (QueueSize is most) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes6) (1)
7	If (Workload is easiest) and (QueueSize is great) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes7) (1)
8	If (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes8) (1)
9	If (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes9) (1)
10	If (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes10) (1)
11	If (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (UserSensitivity is Low) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes11) (1)
12	If (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (UserSensitivity is Low) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes11) (1)
13	If (QueueSize is most) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes13) (1)
14	If (QueueSize is most) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes14) (1)
15	If (QueueSize is most) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes15) (1)
16	If (QueueSize is most) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (UserSensitivity is Low) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes16) (1)
17	If (QueueSize is most) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (JumperStateNo5 is BouncingUp) and (UserSensitivity is Low) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes17) (1)
18	If (QueueSize is great) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes18) (1)
19	If (QueueSize is great) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes19) (1)
20	If (QueueSize is great) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes20) (1)
21	If (QueueSize is great) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (UserSensitivity is Low) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes21) (1)
22	If (QueueSize is great) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (JumperStateNo5 is BouncingUp) and (UserSensitivity is Low) and (PreviousTimeStepMode is InterruptionTask) then (interruptDecision is yes22) (1)
23	If (JumperStateNo1 is FallingDown) and (UserSensitivity is Low) and (PreviousTimeStepMode is InGame) then (interruptDecision is no1) (1)
24	If (JumperStateNo1 is FallingDown) and (JumperStateNo2 is FallingDown) and (UserSensitivity is Low) and (PreviousTimeStepMode is InGame) then (interruptDecision is no2) (1)
25	If (JumperStateNo1 is FallingDown) and (JumperStateNo2 is FallingDown) and (JumperStateNo3 is FallingDown) and (UserSensitivity is Low) and (PreviousTimeStepMode is InGame) then (interruptDecision is no3) (1)
26	If (JumperStateNo1 is FallingDown) and (JumperStateNo2 is FallingDown) and (JumperStateNo3 is FallingDown) and (JumperStateNo4 is FallingDown) and (UserSensitivity is Low) and (PreviousTimeStepMode is InGame) then (interruptDecision is no4) (1)
27	If (JumperStateNo1 is FallingDown) and (JumperStateNo2 is FallingDown) and (JumperStateNo3 is FallingDown) and (JumperStateNo4 is FallingDown) and (JumperStateNo5 is FallingDown) and (UserSensitivity is Low) and (PreviousTimeStepMode is InGame) then (interruptDecision is no5) (1)
28	If (Workload is hard) and (PreviousTimeStepMode is InGame) then (InterruptDecision is no6) (1)
29	If (Workload is hardest) and (PreviousTimeStepMode is InGame) then (InterruptDecision is no7) (1)

Table 32
ANFIS fuzzy rules—all rules model except PreviousTimeStepMode input variable.

Rule #	Rule description
1	If (Workload is easiest) and (QueueSize is small) then (interruptDecision is yes1) (1)
2	If (Workload is easiest) and (QueueSize is small) and (JumperStateNo1 is BouncingUp) then (interruptDecision is yes2) (1)
3	If (Workload is easy) and (QueueSize is small) then (interruptDecision is yes3) (1)
4	If (Workload is easy) and (QueueSize is small) and (JumperStateNo1 is BouncingUp) then (interruptDecision is yes4) (1)
5	If (Workload is easy) and (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) then (interruptDecision is yes5) (1)
6	If (Workload is easiest) and (QueueSize is most) then (interruptDecision is yes6) (1)
7	If (Workload is easiest) and (QueueSize is great) then (interruptDecision is yes7) (1)
8	If (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) then (interruptDecision is yes8) (1)
9	If (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) then (interruptDecision is yes9) (1)
10	If (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) then (interruptDecision is yes10) (1)
11	If (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (UserSensitivity is Low) then (interruptDecision is yes11) (1)
12	If (QueueSize is small) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (UserSensitivity is Low) then (interruptDecision is yes11) (1)
13	If (QueueSize is most) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) then (interruptDecision is yes13) (1)
14	If (QueueSize is most) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) then (interruptDecision is yes14) (1)
15	If (QueueSize is most) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) then (interruptDecision is yes15) (1)
16	If (QueueSize is most) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (UserSensitivity is Low) then (interruptDecision is yes16) (1)
17	If (QueueSize is most) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (JumperStateNo5 is BouncingUp) and (UserSensitivity is Low) then (interruptDecision is yes17) (1)
18	If (QueueSize is great) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) then (interruptDecision is yes18) (1)
19	If (QueueSize is great) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) then (interruptDecision is yes19) (1)
20	If (QueueSize is great) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) then (interruptDecision is yes20) (1)
21	If (QueueSize is great) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (UserSensitivity is Low) then (interruptDecision is yes21) (1)
22	If (QueueSize is great) and (JumperStateNo1 is BouncingUp) and (JumperStateNo2 is BouncingUp) and (JumperStateNo3 is BouncingUp) and (JumperStateNo4 is BouncingUp) and (JumperStateNo5 is BouncingUp) and (UserSensitivity is Low) then (interruptDecision is yes22) (1)
23	If (JumperStateNo1 is FallingDown) and (UserSensitivity is Low) then (interruptDecision is no1) (1)
24	If (JumperStateNo1 is FallingDown) and (JumperStateNo2 is FallingDown) and (UserSensitivity is Low) then (interruptDecision is no2) (1)
25	If (JumperStateNo1 is FallingDown) and (JumperStateNo2 is FallingDown) and (JumperStateNo3 is FallingDown) and (UserSensitivity is Low) then (interruptDecision is no3) (1)
26	If (JumperStateNo1 is FallingDown) and (JumperStateNo2 is FallingDown) and (JumperStateNo3 is FallingDown) and (JumperStateNo4 is FallingDown) and (UserSensitivity is Low) then (interruptDecision is no4) (1)
27	If (JumperStateNo1 is FallingDown) and (JumperStateNo2 is FallingDown) and (JumperStateNo3 is FallingDown) and (JumperStateNo4 is FallingDown) and (JumperStateNo5 is FallingDown) and (UserSensitivity is Low) then (interruptDecision is no5) (1)
28	If (Workload is hard) then (InterruptDecision is no6) (1)
29	If (Workload is hardest) then (InterruptDecision is no7) (1)

Appendix F: functional description of the interruption classifier

The Interruption Classifier is based on an Adaptive Neuro-Fuzzy Inference Systems machine learning tool. The ANFIS is an advanced hybrid soft computing tool that use fuzzy logic and artificial neural networks (Jang et al., 1997). Fuzzy systems and neural networks are two tools that nicely complement one another. While fuzzy logic allows a problem to be viewed on higher and more human-intuitive level, neural networks have been shown to work very effectively in learning, adapting and dealing with raw data (Jang et al., 1997). However, fuzzy systems lack the ability to learn and make self-adjustments. Thus, the amalgamation of a fuzzy system with a neural network into one system (i.e., ANFIS) offered a number of benefits for building the Interruption Classifier. Fig. 7 shows a 2-input ANFIS.

Layer 1 is the input layer. Layer 2 is the fuzzification layer where neurons determine the degree of membership based on the input and quantifier (this is the *if-part* of the fuzzy rules). Layer 3 is the rule layer: each neuron in this layer corresponds to a fuzzy rule. Layer 4 is the normalization layer. Layer 5 performs defuzzification during which each neuron calculates the weighted consequent value of a given rule (this is the *then-part* of the fuzzy rules). Layer 6 produces the final output. For a comprehensive description of ANFIS please see Jang et al. (1997).

ANFIS's offer efficient and effective learning and adaptiveness capabilities (Jang et al., 1997). This is accomplished through forward and backward propagation of error signals through its network. In the forward pass, a training set is presented to the ANFIS, neuron outputs are computed layer-by-layer in the network and the rule-consequent parameters are determined by the least-squares estimator (Jang et al., 1997). These results are then used in the next pass in the learning process—the backward pass. In the backward pass the back-propagation algorithm is used. Error signals are sent backwards through the network and the antecedent parameters are tuned appropriately using the chain rule (Jang et al., 1997). In this way, an ANFIS can learn and adapt quickly based on training datasets. In this work, the successful models created by the Classifier were based on the following principles:

Defining fuzzy rules and membership functions: The success of a model is dependent on well-defined rules that represent the task and user contexts as fully and precisely as possible. A set of rules were created based on user modeling information such as personality traits, frustration level, tolerance to interruptions, etc. Additional set of rules characterized task details in combination with the user's real-time activities. For example, in the main experiment, the fuzzy variable, *Workload*, is dynamically adjusted based on the number of jumpers in the primary (game) task. This variable was characterised by 4 membership functions: *easiest, easy, hard, hardest*. Similarly, *QueueSize* representing the number waiting of interruption tasks,

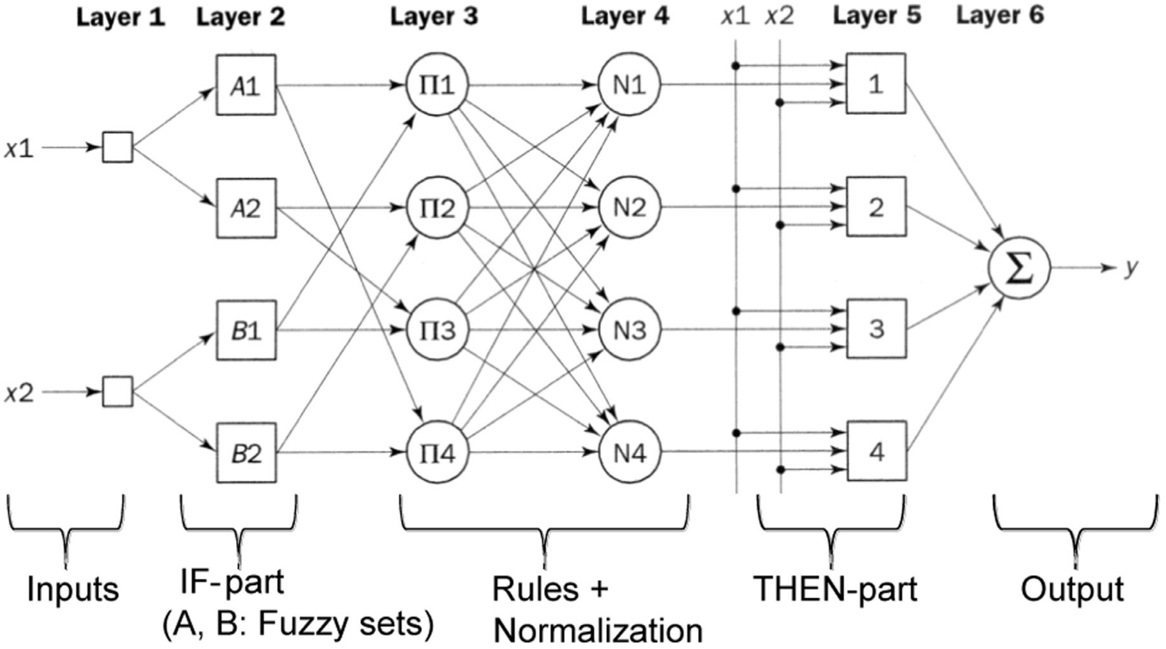


Fig. 7. 2-input adaptive Neuro-Fuzzy inference system.

was characterized by 4 membership functions: *least, small, great* and *most*. Each membership function was a trapezoidal wave form carefully constructed from ANFIS research literature, researcher observation, user's strategies and empirical evidence. The Classifier then tuned these functions based on the training datasets.

Training datasets: Using different training datasets had a profound impact on the models created. Several participant datasets produced models that met and exceeded the performance criteria. These models performed well and were generalizable across the entire participant group. Analysis showed that these participants were very consistent in their strategies and interruption timings when performing the tasks.

Checking datasets: Checking datasets were used for model validation. The model error for the checking data set tends to decrease as the training takes place up to the point that overfitting begins (Jang, 1996). A substantial amount of research was conducted to reduce overfitting. This was accomplished by experimenting with different checking datasets and the number of epochs (iterations of forward-backward passes) for the ANFIS to learn effectively.

Historic event knowledge: The impact of historic event knowledge is significant. Knowing the past, especially when it is 100 ms or less in the past has a profound impact in reasoning and deciding whether to interrupt the user at a given point in time. All models created that used historic knowledge performed extremely well.

The Classifier is generalizable to a degree. We discovered that user modeling information such as personality traits are generalizable across all models in all the experiments conducted. However, other aspects of the modeling process (e.g., task details), need to be explicitly represented in a model for it to be successful and are therefore not directly generalizable.

Appendix G: raw numbers for the cases of ``interruptions" relative to the total number of cases

Participant #	% Cases correctly classified	total # of cases (both "interrupt" and "do not interrupt") (ms)	Time in interruption cases (ms)	% of interruption cases / total # of cases
1	98.799	270,000.000	55,727.924	20.640
2	99.795	270,000.000	46,340.689	17.163
3	97.372	270,000.000	43,297.927	16.036
4	99.839	270,000.000	48,495.032	17.961
5	98.726	270,000.000	68,743.174	25.460
6	99.202	270,000.000	52,578.077	19.473
7	98.112	270,000.000	42,764.913	15.839
8	94.994	270,000.000	47,395.328	17.554
9	99.846	270,000.000	48,066.241	17.802
10	99.649	270,000.000	47,859.298	17.726
11	99.180	270,000.000	41,405.168	15.335
12	97.848	270,000.000	64,242.230	23.793
13	99.663	270,000.000	79,478.079	29.436
14	98.690	270,000.000	60,126.867	22.269
15	98.316	270,000.000	57,187.750	21.181
16	99.649	270,000.000	60,934.798	22.568
17	99.407	270,000.000	58,854.531	21.798

18	99.414	270,000.000	62,066.451	22.988
19	99.517	270,000.000	46,867.749	17.358
20	98.763	270,000.000	59,503.997	22.039
21	99.810	270,000.000	51,865.072	19.209
22	99.876	270,000.000	52,997.285	19.629
23	99.810	270,000.000	29,388.352	10.885
24	99.392	270,000.000	73,065.327	27.061
25	99.597	270,000.000	30,644.713	11.350
Min	94.994		29,388.352	10.885
Max	99.876		79,478.079	29.436
Mean	99.011		53,195.879	19.702
Median	**99.407**		**52,578.077**	**19.473**
Std Dev	1.061		11,577.532	4.288

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ijhcs.2018.06.005.

References

Altmann, E.M., Trafton, J.G., 2002. Memory for goals: an activation-based model. Cognit. Sci. 26 (1), 39–83.

Altmann, E.M., Trafton, J.G., Hambrick, D.Z., 2014. Momentary interruptions can derail the train of thought. J. Exp. Psychol. 143 (1), 215–226.

Baethge, A., Rigotti, T., Roe, R.A., 2014. Just more of the same, or different? An integrative theoretical framework for the study of cumulative interruptions at work. Eur. J. Work Org. Psychol. 23 (1).

Bower, G.H., Morrow, D.G., 1990. Mental models in narrative comprehension. Science 247 (4938), 44–49.

Davison, A.K., Lansley, C., Costen, N., Tan, K., Yap, M.H., 2018. SAMM: a spontaneous micro-facial movement dataset. IEEE Trans. Affect. Comput. 9 (1), 116–129.

Dobrian, F., Sekar, V., Awan, A., Stoica, I., Joseph, D., Ganjam, A., . . . Zhang, H., 2011. Understanding the impact of video quality on user engagement. Paper presented at the In: Proceedings of the ACM SIGCOMM 2011 Conference. Toronto, Ontario, Canada.

Elkan, C., 2011. Evaluating Classifiers. Retrieved from. http://cseweb.ucsd.edu/~elkan/250B/classifiereval.pdf.

Finn, A., Limerick, N., 2003. Active learning selection strategies for information extraction. In: Paper presented at the Proceedings of the International Workshop on Adaptive Text Extraction and Mining. Croatia.

Fogarty, J., Ko, A.J., Aung, H.H., Golden, E., Tang, K.P., Hudson, S.,E., 2005. Examining task engagement in sensor-based statistical models of human interruptibility. In: Paper presented at the Conference on Human Factors in Computing Systems. Portland, OR.

Forman, G., Scholz, M., 2010. Apples to apples in cross-validation studies: pitfalls in classifier performance measurement. ACM SIGKDD Explor. 12 (1), 49–57.

Frías-Martínez, E., Magoulas, G., Chen, S., Macredie, R., 2004. Recent soft computing approaches to user modeling in adaptive hypermedia. Adapt. Hypermed. 3137, 104–114.

Garcıa, J.C.F., Mendez, J.J.S., 2007. A Comparison of ANFIS, ANN and DBR systems on volatile time series identification. In: Paper presented at the Annual Meeting of the North American Fuzzy Information Processing Society.

Gharaviri, A., Dehghan, F., Teshnelab, M., Abrishami, H.M., 2008. Comparison of neural networks, ANFIS, and SVM classifiers for PVC arrhythmia detection. In: Paper presented at the Proceedings of the Seventh International Conference on Mach. Learn. and Cybernetics, Kunming.

Gievska, S., Lindeman, R., Sibert, J., 2005. Examining the Qualitative Gains of Mediating Human Interruptions during HCI. Paper presented at the In: HCII. Las Vegas, Nevada.

Gievska, S., Sibert, J., 2004. Empirical validation of a computer-mediated coordination of interruption. In: Paper presented at the CHISIG - OZCHI 2004: Supporting Community Interaction. Wollongong, Australia.

Gievska, S., Sibert, J., 2005. Using task context variables for selecting the best timing for interrupting users. Paper presented at the In: Smart Objects and Ambient Intelligence Conference. Grenoble, France.

Gluck, J., Bunt, A., McGrenere, J., 2007. Matching attentional draw with utility in interruption. Paper presented at the In: CHI 2007 Attention & Interruption. San Jose, CA, USA.

Guinn, C.I., 1999. Evaluating mixed-initiative dialog. IEEE Intell. Syst. 14 (5), 21–23.

Hamilton, H.J., 2011. Confusion Matrix. Retrieved from. http://www2.cs.uregina.ca/~hamilton/courses/831/notes/confusion_matrix/confusion_matrix.html.

Hancock, P.A., Meshkati, N., 1988. Human Mental Workload. Elsevier Science.

Hertzum, M., Holmegaard, K.D., 2013. Perceived time as a measure of mental workload: effects of time constraints and task success. Int. J. Human Comput. Interact. 29 (1), 26–39.

Hodgetts, H.M., Jones, D.M., 2003. Interruptions in the tower of London task: can preparation minimise disruption? In: Paper presented at the Human Factors and Ergonomics Society 47th Annual Meeting. Denver, Colorado.

Hodgetts, H.M., Jones, D.M., 2007. Reminders, alerts and pop-ups: the cost of computer-initiated interruptions. In: Jacko, J. (Ed.), Hum. Comput. Interact. Springer-Verlag, Berlin Heidelberg, pp. 818–826.

Horvitz, E., Kadie, C., Paek, T., Hovel, D., 2003. Models of attention in computing and communication from principles to applications. Commun. ACM 46 (3), 52–59.

Horvitz, E., Koch, P., Apacible, J., 2004. BusyBody: creating and fielding personalized models of the cost of interruption. Paper presented at the In: CSCW'04. Chicago, IL.

Horvitz, E., Oliver, N., 2005. A comparison of HMMs and dynamic bayesian networks for recognizing office activities. Paper presented at the User Modeling 2005 Edinburgh In: Edinburgh. http://research.microsoft.com/~horvitz/DBN_HMM.pdf.

Iqbal, S., Bailey, B., 2005. Paper presented at the In: CHI 2005. Portland, Oregon, USA. Paper presented at the.

Iqbal, S., Bailey, B., 2006. Paper presented at the In: CHI 2006. Montreal, Quebec, Canada. Paper presented at the.

Iqbal, S., Bailey, B., 2007. Paper presented at the In: CHI 2007. San Jose, California, USA. Paper presented at the.

Iqbal, S., Horvitz, E., 2010. Notifications and awareness: a field study of alert usage and preferences. In: Paper presented at the 2010 ACM conference on Computer supported cooperative work. New York, NY, USA.

Iqbal, S.,T., Bailey, B.,P., 2008. Effects of intelligent notification management on users and their tasks. Paper presented at the In: SIGCHI Conference on Human Factors in Computing Systems. New York, NY, USA.

Iqbal, S.T., Bailey, B.P., 2010. Oasis: A framework for linking notification delivery to the perceptual structure of goal-directed tasks. ACM Trans. Comput. Hum. Interact. 17 (4), 1–28.

Jang, J.-S.R., Sun, C.-T., Mizutani, E., 1997. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice Hall, Englewood Cliffs, NJ.

Jang, J.-S.R., 1996. Input Selection for ANFIS Learning. Paper presented at the Fifth IEEE International Conference on Fuzzy Systems. New Orleans, LA, USA.

Jensen, A.R., Rohwer, W.D.Jr, 1966. The Stroop color-word test: a review. Acta Psychologica 25, 36–93.

Kohavi, R., Provost, F., 1998. Glossary of terms: special issue on applications of machine learning and the knowledge discovery process. Mach. Learn. 30, 271–274.

Li, Y., Fuller, B., 2017. “I'm Lovin' IT”: toward a technophilia model of user adaptation to ICT. Paper presented at the In: Advances in Management Information Systems Research.

Lisetti, C.L., Nasoz, F., 2004. Using noninvasivewearable computers to recognize human emotions from physiological signals. J. Appl. Sig. Process. 11, 1672–1687.

Loukopoulos, L.D., Dismukes, R.K., Barshi, I., 2009. The Multitasking Myth: Handling Complexity in Real-World Operations. Routledge.

Lu, Z., Szafron, D., Greiner, R., Lu, R., Wishart, D.S., Poulin, B., Eisner, R., 2004. Predicting subcellular localization of proteins using machine-learned classifiers. Bioinformatics 20 (4), 547–556.

Lucassen, G., Dalpiaz, F.E.M., van der Werf, J.M., Brinkkemper, S., 2017. Improving user story practice with the grimm method: a multiple case study in the software industry. Requirements Engineering: Foundation for Software Quality 10153 *vol*.

McFarlane, D.C., Latorella, K.A., 2002. The scope and importance of human interruption in HCI design. Hum. Comput. Interact 17, 1–61.

Negnevitsky, M., 2004. Artificial Intelligence: A Guide to Intelligent Systems. Addison-Wesley.

Picard, R.W., 2003. Applications of affective computing in hum. comput. interact. Int. J. Hum. Comput. Stud. 59 (1-2), 55–64.

Pielot, M., Cardoso, B., Katevas, K., Serr, J., #224, Matic, A., Oliver, N., 2017. Beyond interruptibility: predicting opportune moments to engage mobile phone users. In: Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. 1. pp. 1–25. https://doi.org/10.1145/3130956.

Pielot, M., Church, K., Oliveira, R.d., 2014. An in-situ study of mobile phone notifications. Paper presented at the In: Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services. Toronto, ON, Canada.

David, Prabu, Kim, Jung-Hyun, Brickman, JaredS, Ran, Weina, Curtis, C.M., 2015. Mobile phone distraction while studying. New Media Soc. 17 (10).

Pu, P., Viappiani, P., Faltings, B., 2006. Increasing user decision accuracy using suggestions. Paper presented at the In: CHI 2006. Montreal, Quebec, Canada.

Reisenzein, R., Horstmann, G., Schützwohl, A., 2017. The cognitive evolutionary model of surprise: a review of the evidence. Top. Cognit. Sci.

Rind, A., Wang, T.D., Aigner, W., Miksch, S., Wongsuphasawat, K., Plaisant, C., Shneiderman, B., 2011. Interactive information visualization to explore and query electronic health records. Found. Trends Hum. Comput. Interact. 5 (3), 207–298.

Sarker, I.H., Kabir, M.A., Colman, A., Han, J., 2017. Designing architecture of a rule-based system for managing phone call interruptions. In: Paper presented at the Proceedings of the 2017 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers. Maui, Hawaii.

Scheirer, J., Fernandez, R., Klein, J., Picard, R.W., 2002. Frustrating the user on purpose: a step toward building an affective computer. Interact. Comput. 14, 93–118.

Solingen, R., Berghout, E., Latum, F., 1998. Interrupts: just a minute never is. IEEE Softw. 15, 97–103.

Tulga, M.K., Sheridan, T.B., 1980. Dynamic decisions and work load in multitask supervisory control. IEEE Trans. Syst. Man Cybern. 10 (5), 217–232.

Warm, J.S., Dember, W.N., Hancock, P.,A., 1996. Vigilance and workload in automated systems. In: Parasuraman, R., Mouloua, M. (Eds.), Automation and Human Performance: Theory and Applications. Lawrence Erlbaum.