Graphical Management of Attentional State

by

Sean William Kortschot

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy

Mechanical & Industrial Engineering
University of Toronto

Graphical Management of Attentional State

Sean William Kortschot

Doctor of Philosophy

Mechanical & Industrial Engineering
University of Toronto

2019

Abstract

Adaptive user interfaces (AUIs) modify their information content, structure, or interactions in response to changes in operating context, called *triggers*. This dissertation focuses on the operator measurement class of triggers, which initiate interface adaptations based on changes to an operator's physical or mental state. To infer a user's mental state, we used Passive Data Monitoring (PDM), which is a method that examines streams of interaction data for patterns that are characteristic of an attentional state. AUIs can then search for these patterns and adapt an interface accordingly. This dissertation presents three studies that advance the methodology by which attentional states are detected via PDM and responded to via AUIs.

The first study describes two experiments, which identified the behavioural correlates of attentional switches and implemented interventions that mitigated the performance decrement incurred from those switches, respectively. Experiment 1.1 revealed that there was a significant interruption in participant interaction both before and after attentional switches. Experiment 1.2 showed that both of these delays could be reduced by implementing an intervention that contextualized a switch.

The second study describes one experiment aimed at developing a machine learning classifier that could detect when participants were in an attentional tunnel. A Convolutional Long Short-Term Memory neural network accurately identified states of attentional tunneling significantly above chance level. This experiment also explored the performance tradeoffs stemming from attentional tunnels.

The third study describes two experiments aimed at developing machine learning classifiers capable of detecting information overload and using those classifiers as the engine for an AUI

that performed online calibration of the degree of concurrent information presentation, respectively. In the first experiment, four classifiers were developed that all performed above chance level. In the second experiment, two of these classifiers drove two separate AUIs that significantly benefited some user groups relative to experimental controls. The experiment revealed significant differences in how user groups responded to AUIs.

Overall, this dissertation contributes to our understanding of how attentional states manifest behaviourally. It also formalizes PDM as a method for understanding, detecting, and leveraging these behavioural manifestations for use in AUIs.

Acknowledgments

First, let me thank my parents and Phoebe for your love and support throughout my studies. Thank you as well to Chris, Emily and the rest of my family.

I would like to thank Greg Jamieson for providing guidance and friendship over the years that I've spent in CEL. Your feedback and mentorship has helped to shape the way that I approach problems both in research and in life.

Thank you to Michael Dorneich for providing insightful feedback and for lending your expertise to this dissertation.

I would also like to thank Birsen Donmez and Jonathan Cant for your thoughtful guidance over the course of my dissertation. Your feedback has guided this research and it wouldn't have been possible without you.

I had the opportunity to work alongside wonderful lab mates during my time at CEL, so thank you to all of you.

Thank you as well to my co-authors, Dusan Sovilj, Chelsea Carrasco, Harold Soh, Alexis Morris, Scott Sanner, and Amrit Prasad.

Table of Contents

List of Tables

List of Figures

Chapter 1 - Introduction

"One of the main results of Twentieth-century Cognitive Psychology is that, despite the overall impressive abilities of people to sense, remember, and reason about the world, our cognitive abilities are extremely limited in well-characterized ways."

Horvitz, Kadie, Paek, & Hovel (2003)

The amount of information that is available to operators has significantly grown in recent years. With sensors becoming increasingly accessible, reliable, and networked (E. A. Lee, 2008; Morales-Herrera, Fernández-Caballero, Somolinos, & Sira-Ramírez, 2017), this trend is likely to continue. The expansion of information availability has resulted in increased cognitive demands associated with monitoring and understanding that information (Woods & Patterson, 2000). Operators therefore rely heavily on decision support systems (DSSs), whose principal goal is to facilitate the extraction of meaningful insights from the copious information that is typically available to them. Although DSSs have become integral to monitoring and controlling virtually all complex systems, the information load imposed on operators remains significant. This presents an opportunity for adaptive user interfaces (AUIs), which modify their information content, configuration, or interactions in response to changes in operating context (Feigh, Dorneich, & Hayes, 2012).

AUIs are predicated on the assumption that different contexts impose different constraints on an interface. For example, in particularly taxing operating contexts, interfaces that present less information may be better suited than more information-rich displays (Dorneich, Whitlow, Mathan, Carciofini, & Ververs, 2005). Although the motivation and justification for AUIs is clear, and although they have been referenced in literature as a promising design strategy for nearly five decades (e.g., Enstrom & Rouse, 1977), they have yet to gain significant traction outside of experimental settings. This difficulty has been attributed to foundational limitations of adaptive systems (Bainbridge, 1983; Strauch, 2018). However, recent advances in machine learning may offer new avenues to overcoming some of limitations that have hindered past adaptive systems. More specifically, these advances may enable a significantly more

comprehensive understanding of the contextual changes that AUIs seek to address (Feigh et al., 2012; Mangos & Hulse, 2017).

Irrespective of the computational capabilities of an adaptive system, there are three challenges that must be overcome in order for an AUI to successfully perform adaptations (Rothrock, Koubek, Fuchs, Haas, & Salvendy, 2002): 1) Identifying what to respond to, 2) Understanding when to respond, and 3) Understanding how to respond. The following sections will outline the approach that this dissertation takes to address these challenges. Additional literature is presented in Appendix A.

1.1. What to respond to

The first step in developing an AUI is identifying what should *trigger* the adaptation. Feigh et al. (2012) list five classes of triggers: Operator measurement, system state, spatio/temporal, task/mission, or environmental. Changes in any of these five classes can initiate adaptations. For example, a spatio/temporal trigger for an AUI comes in the form of location- and time-based personalized recommendation systems that may recommend a particular smart phone app depending on where you are located or when you are using your phone (e.g., Hosub Lee, Young Sang Choi, & Yeo-Jin Kim, 2011). A common example of an environmentally attuned adaptive feature is *night-mode*, wherein the brightness or hue of an interface responds to the ambient light in the environment (e.g., Nagare, Plitnick, & Figueiro, 2019). While these trigger classes have yielded a diverse set of real-world applications, the operator measurement class of triggers has provided fewer successful applications (Feigh et al., 2012). Operator measurement triggers initiate adaptations in response to changes in an operator's physical or mental state. This class assumes that different operator states imply different information or interaction needs. For example, operators who are experiencing high workload tend to narrow their attentional focus (Rantanen & Goldberg, 1999). This narrowed attentional scope influences how an operator perceives information and is therefore also likely to influence how that information should be presented.

Responding to an operator's state begins with detecting that state. One method described in Feigh et al. (2012) for inferring operator state is via performance-based measures. This can either be from the number of errors recorded (e.g., Virvou, 1999), or through some other real-time performance metric (e.g., temporal; Benyon & Murray, 1993). In some domains (e.g., driving)

where online performance metrics are available (e.g., lane deviation), this approach for operator state inference is viable (e.g., drowsiness; Forsman, Vila, Short, Mott, & Van Dongen, 2013). However, the majority of real-world domains lack a continuous and accurate performance metric. This lack has led the majority of research to focus on the second method described in Feigh et al. (2012): operator physiology, which derives cognitive state inferences from operator biometrics. As physiological sensors have become increasingly accessible and reliable (McLane et al., 2015; Mukhopadhyay & Lay-Ekuakille, 2010), this approach has garnered increased interest. An example of physiological state inference is the *Communication's Scheduler*, which employs several biometric signals (e.g., electroencephalogram, electrocardiogram) to infer a soldier's workload for adapting the manner by which they receive communications (Dorneich, Whitlow, et al., 2007). Beyond workload (Borghetti, Giametta, & Rusnock, 2017; Wilson & Russell, 2004), physiological relationships have been found with mind wandering (Baldwin et al., 2017), task engagement (Berka et al., 2007), attentional tunneling (Régis et al., 2014), and various emotional states (Y. Y. Lee & Hsieh, 2014).

Although physiological state inference has proven useful in test settings, it faces significant limitations with adoption requiring initial financial investment in physiological sensors and users who are willing to wear those sensors and share their data. This problem is negligible in applied spaces such as military settings where soldiers are already wearing a significant amount of costly equipment. However, in either civilian spaces or in domains where operators work long, sedentary shifts, outfitting them with physiological sensors can be costly and cumbersome. This presents an opportunity for Passive Data Monitoring (PDM), which is a method for inferring an operator's state that exploits streams of behavioural data that exist independent of physiology or task performance (e.g., cursor position) and determines if patterns within that data align with cognitive events or states (Kortschot et al., 2018; Palmius et al., 2016).

There are notable demonstrations of the efficacy of PDM. McDonald, Lee, Schwarz, and Brown (2014) tested whether states of drowsiness could be detected in drivers by examining their interactions with the steering wheel. They achieved comparable accuracy to the standard physiological measure (Percentage Eye Closure) for assessing drowsiness, but without the need for a camera setup. Palmius et al. (2016) tested whether GPS data could be used to detect impending states of depression in a sample of participants with Bipolar Depression. They found participants' geographical movements tended to concentrate around their home as they

approached a depressive episode. They used this method to detect oncoming depressive episodes with an accuracy of about 85%, which is comparable to equivalent physiological measures (Valenza et al., 2015). As both McDonald et al. (2014) and Palmius et al. (2016) demonstrate, successful detection of operator state can be achieved without the use of any additional sensors, secondary tasks, or performance metrics. This suggests the viability of PDM as either an alternative or augmentation to physiological measurement for operator state inference.

1.2. When to respond

Once the type of signal used for a trigger is identified, an AUI must determine when to respond to that signal. Although future AUIs may be attuned to a spectrum of operator states, useful test cases at this stage of AUI development tend to focus on negative states requiring an intervention. A useful class of these negative states are cognitive bottlenecks, which are resource-limitations in human cognition that hinder performance (Besner, Reynolds, & O'Malley, 2009). They arise across a breadth of psychological faculties including perception (Salvucci & Taatgen, 2008), response selection (Nobre & Kastner, 2014), and memory (Borst, Taatgen, & van Rijn, 2010). Typically, bottlenecks arise in scenarios where cognitive demands exceeds cognitive resources, often as a result of multitasking or information overload (Borst et al., 2010). Tasks involving these characteristics are ubiquitous in both professional and civilian domains.

Dorneich et al. (Dorneich, Whitlow, Ververs, & Rogers, 2003) outline a framework for describing cognitive bottlenecks as supply and demand imbalances. In it, they describe three types of environments that are likely to impose resource-imbalance-bottlenecks on operators. The first, information rich environments, include domains wherein the amount of information available to an operator exceeds the amount of information they are capable of processing. The second, dynamic environments, are domains wherein the rate of information change exceeds the rate that an operator is capable of processing. The third is environments wherein multitasking is required such that the number of tasks that need to be performed exceed the attentional or cognitive resources of the operators. The impact of any one of these characteristics depends heavily on the levels of the other two. For example, the rate of change that an operator is capable of perceiving depends heavily on the amount of information that is changing. Similarly, if an operator is required to perform multiple tasks, their ability to perceive large amounts of data pertaining to any one of those tasks is limited (Katidioti & Taatgen, 2014). An AUI operating in

an environment that has these conditions should therefore be able to listen for signals that are indicative of a cognitive bottleneck and institute interventions aimed at mitigating the negative effects of that bottleneck.

1.3. How to respond

The exact nature of how an AUI responds to a cognitive bottleneck should be contingent on the characteristics of that bottleneck. For example, if a bottleneck is particularly relevant to the amount of information presented to a user (e.g., information overload), then adapting the amount of information is a logical first step in designing an AUI. Feigh et al. (2012) present a taxonomy of adaptations that includes four principal categories of adaptation: Modification of Functional Allocation, Modification of Task Scheduling, Modification of Interaction, and Modification of Content. The scope of this dissertation is limited to modifications of interaction and content, although it is likely that there are AUI designs that can treat cognitive bottlenecks via modifications to functional allocation or task scheduling.

The Communications Scheduler (Dorneich, Whitlow, et al., 2007) performs adaptations that can be classified under the Modification of Interaction branch of the Feigh et al. (2012) taxonomy. This system identifies periods of high workload and modifies the manner by which soldiers access communications, shifting from a push to pull interaction. This intervention mechanism is tailored to the specific characteristics of the bottleneck that they are treating as well as the domain in which that bottleneck is occurring. It gives soldiers freedom to access communications with the understanding that high workload may be indicative of potentially life-threatening scenarios. In a different, non-safety critical environment, this intervention strategy may not be as well-suited.

AUIs that modify the information content are relatively common. For example, Grawemeyer and Cox (2005) developed an adaptive system that tailored the format of information presentation to the specific needs of the user. This did not modify the content that was displayed, just the manner of presentation. Similarly, the adaptive version of TaskSieve (Ahn & Brusilovsky, 2013) employed a user model to proactively adapt the manner in which details during an information retrieval task were displayed. Again, these interventions are based on the domains in which they are operating.

1.4. Research Outline

This dissertation is divided into three phases, each examining a different cognitive bottleneck in relation to AUIs. There are two types of experiments in this dissertation. *Discovery* experiments aim to identify the behavioural correlates of a cognitive bottleneck. *Intervention* experiments aim to introduce an intervention to mitigate the negative effects of a bottleneck.

Each chapter presents a journal paper that describes a different cognitive bottleneck and advances the methodology by which that bottleneck is detected and treated. The first paper (Kortschot et al., 2018) presents two experiments (Experiment 1.1 & 1.2). Experiment 1.1 presented users with a cybersecurity microworld simulation and sought to identify behavioural markers of *attentional switches.* Experiment 1.2 introduced an intervention strategy aimed at mitigating the performance decrement incurred from these attentional switches. The second paper (Kortschot & Jamieson, in press) presents one experiment (Experiment 2.1), which aimed to induce a state of *attentional tunneling* in participants and subsequently developed a machine learning classifier that could identify the behavioural markers of that state relative to a baseline. The third paper (Kortschot & Jamieson, in review) describes two experiments (Experiment 3.1 & 3.2), aimed respectively at identifying the behavioural correlates of *information overload* with machine learning classifiers and using these classifiers as the engine for an AUI that calibrated the amount of information that was concurrently presented to users. These papers span two parallel and overlapping progressions. The first advances from detecting attentional events to detecting attentional states. The second progression moves from detecting and intervening in operator state via an offline, heuristic approach to an online approach that detects attentional state in real time and adapts the interface accordingly. Figure 1 presents an overview of the three phases of this dissertation.

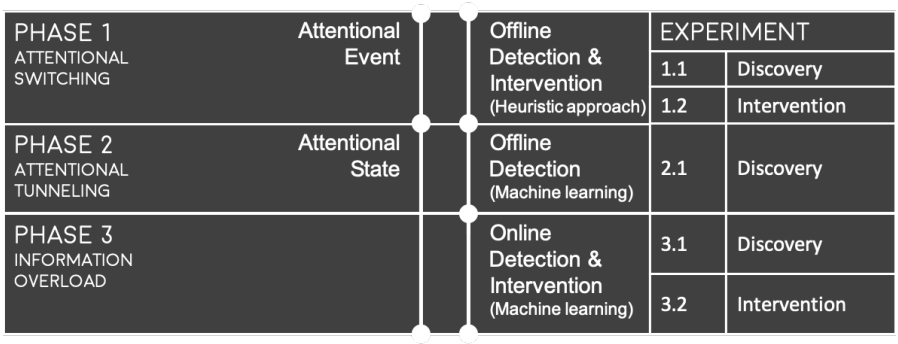


Figure 1. Summary of research approach.

Chapter 2 - Attentional Switching

The first phase of this dissertation comprised Experiments 1.1 and 1.2 (see Figure 1). The goal of Experiment 1.1 was to establish whether performance decrements stemming from attentional switches could be detected through behavioural indices. This experiment largely served as proof of concept for using behaviour to detect attentional states (i.e., PDM). Upon successfully developing a methodology by which the performance decrements stemming from attentional switches could be detected, Experiment 1.2 then sought to develop interventions that reduced the negative impact of attentional switches. Using the methods developed in Experiment 1.1, the efficacy of these intervention strategies was assessed.

2.1. Statement of Authorship

This chapter was published in Human Factors (Kortschot et al., 2018) and was authored by Sean W. Kortschot, Dusan Sovilj, Greg A. Jamieson, Scott Sanner, Chelsea Carrasco, and Harold Soh. The research questions and statistical methodology (i.e., PDM) were conceptualized, developed, and defined by Sean W. Kortschot under the supervision of Greg A. Jamieson. The literature search, research design, selection of measurements, and the selection, execution, and interpretation of statistical tests were all conducted by Sean W. Kortschot. Both experiments described herein were conducted using an experimental testbed called the Adaptive Network Visualization for Experimental Learning (ANVEL). The front-end design of ANVEL was performed by Sean W. Kortschot and Chelsea Carrasco. Dusan Sovilj, Scott Sanner, and Harold Soh were principally responsible for the back-end development of ANVEL. This included development of the interface as well as the machine learning algorithms that drove a system that recommended areas of a network graph to attend to. Sean W. Kortschot designed the interventions used in the second experiment. He also conducted the experimentation for all participants in both experiments. Finally, this chapter was written and revised by Sean W. Kortschot with Greg A. Jamieson providing editorial oversight throughout both the writing and revision processes. All other authors reviewed and commented on the content prior to submission and publication.

Measuring and Mitigating the Costs of Attentional Switches in Active Network Monitoring for Cyber Security

2.2. Abstract

Objective: Characterize the behavioural costs of attentional switches between points in a network map and assess the efficacy of interventions intended to reduce those costs.

Background: Cyber security network operators are tasked with determining an appropriate attentional allocation scheme given the state of the network, which requires repeated attentional switches. These attentional switches may result in temporal performance decrements, during which operators disengage from one attentional fixation point and engage with another.

Method: We ran two experiments where participants identified a chain of malicious emails within a network. All interactions with the system were logged and analyzed to determine if users experienced disengagement- and engagement-delays.

Results: Both experiments revealed significant costs from attentional switches before (i.e., disengagement) and after (i.e., engagement) participants navigated to a new area in the network. In our second experiment, we found that interventions aimed at contextualizing navigation actions lessened both disengagement- and engagement-delays.

Conclusion: Attentional switches are detrimental to operator performance. Their costs can be reduced by design features that contextualize navigations through an interface.

Application: This research can be applied to the identification and mitigation of attentional switching costs in a variety of visual search tasks. Furthermore, it demonstrates the efficacy of non-invasive behavioural monitoring for inferring cognitive events.

2.3. Introduction

As dependence on networked systems has increased, the global vulnerability to cyber-crime has grown (Goutam, 2015). This trend is mirrored by the growing number of yearly cyber-attacks (Ben-Asher & Gonzalez, 2015), with the net economic cost of data breaches expected to exceed $2 trillion by 2019 (Moar, 2015). A significant investment in cyber-security measures is

therefore necessary for the protection of organizational activities and finances. Despite computational safeguards such as antivirus software and firewalls (Alrajeh, Khan, & Shams, 2013), any sufficiently large network remains vulnerable to attacks (Ahmad, Hadgkiss, & Ruighaver, 2012). In these instances, active network monitoring (ANM) is necessary.

ANM is the process of detecting, diagnosing, and mitigating the effects of network intrusions or attacks (D'Amico, Whitley, Tesone, O'Brien, & Roth, 2005). This task is extremely complex given the magnitude and distribution of the networks (Mitropoulos, Patsos, & Douligeris, 2006), the coordination required between operators (Werlinger, Muldner, Hawkey, & Beznosov, 2010; Tyworth, Giacobe, & Mancuso, 2012), significant time pressure and stakes (Khan, Gani, Abdul Wahab, Shiraz, & Khan, 2016), and the difficulty of determining an appropriate course of action for each unique attack (Werlinger et al., 2010). These challenges necessitate heavy reliance on decision support systems (DSSs). A typical cyber-security DSS detects anomalous or malicious behaviour by comparing both the current overall state and individual activities of the network against both the expected activity under non-attack conditions and against known attacks (Ashfaq & Khayam, 2011). These relationships are then depicted in tables, which often provide *scores* for elements in the network, and graphs that support understanding of patterns of data and alerts.

The meaning of an element's score differs from system to system, but it is generally a function of the probability and the cost of malicious behaviour within that element (Ashfaq & Khayam, 2011). For example, moderately anomalous behaviour in an important server would elicit a higher score than highly anomalous behaviour from a low-level email account. Operators combine scores with network graphs to understand how alerts are distributed to identify functional relationships (Franke & Brynielsson, 2014). Operators then use this higher-order understanding of the system to determine how to allocate their attention (Hopf, Boehler, Schoenfeld, Heinze, & Tsotsos, 2010; Olszewski, 2014; Pfleeger & Caputo, 2012).

Operators are required to repeatedly determine where to focus their attention amongst a multitude of viable options. This near-constant demand for attention allocation, coupled with extreme information quantity is likely to result in *cognitive bottlenecks,* which are performance-limiting constraints in the information flow between the system, the human, and the situation (Dorneich et al., 2003). Bottlenecks have been studied in perception (Salvucci & Taatgen, 2008),

response selection (Nobre & Kastner, 2014), and memory (Borst et al., 2010), as well as across domains like driving (Donmez, Boyle, & Lee, 2006) and command and control of military operations (Dorneich, Mathan, Ververs, & Whitlow, 2007). While several cognitive bottlenecks are likely to be implicated in ANM, we focus on bottlenecks stemming from visual *attentional switching* here. Attentional switching is the process of moving one's focus from one point to another and consists of three phases: disengagement from current fixation point, shifting to a new location, and engagement of new fixation point (Posner & Presti, 1987). We concentrate our analysis on the cognitive costs of the disengagement- and engagement-phases of the attentional switch.

Although engagement and disengagement are separated by the shifting phase, they are closely related. For example, engagement with a new target is hindered if participants need to first disengage from an initial target (Duncan, 1980). Ettwig and Bronkhorst (2015) corroborated this, showing that even when a previously perceived stimulus is masked, switching attention to a new information stream is hindered, a finding explained as a need to disengage from the masked stimuli even after it has disappeared. Dombrowe, Donk, and Olivers (2011) found that eye saccade accuracy and speed were significantly hindered when participants were asked to scan a series of targets of different colours, illustrating the potential performance decrements incurred from attentional switches. Longman, Lavric, & Monsell (2017) showed that allowing participants to prepare for an upcoming switch lessened the cost of switching but did not eliminate it. The common thread in the above research is that there is a clear cost incurred from disengaging from a previous attentional fixation and engaging with a new one. In real-world settings, where a large number of attentional switches are necessary, the cumulative cost of disengaging and engaging with targets could be detrimental to performance. We investigated this relationship in an ANM setting that more closely approximated real-world tasks than did the paradigms used to attain the evidence presented above.

This paper describes two studies, whose respective principal aims were to, 1) examine how attentional switches impact operator behaviour in ANM, and 2) to determine the efficacies of two interventions aimed at mitigating the negative effects of attentional switches.

2.4. Overall Method

2.4.1. Experimental Platform, Participants, and Scenario Design

Our experiments were conducted on a platform that allowed for the representation and visualization of a communication network, provided users with the tools to inspect elements within that network, and allowed users to tag elements as normal, suspicious, or malicious (See Kortschot et al., 2017). The network was populated with the 2015 version of the Enron Email Corpus (https://www.cs.cmu.edu/~./enron/), which is a public dataset of real emails. The visualization of the network was simple, with nodes representing users and edges representing emails. Each email in the network had a score, which was the probability that it contained malicious content. Users could zoom in on areas of the network and inspect an email by either selecting it from a list or directly on the network map. They could also click on users (i.e., nodes) in the network map and view all of their outgoing emails. The Top-K slider displayed zero to ten of the highest scored emails. For example, if the Top-K slider was set to one, it would highlight the single email in the network map with the highest score (see Figure 2).

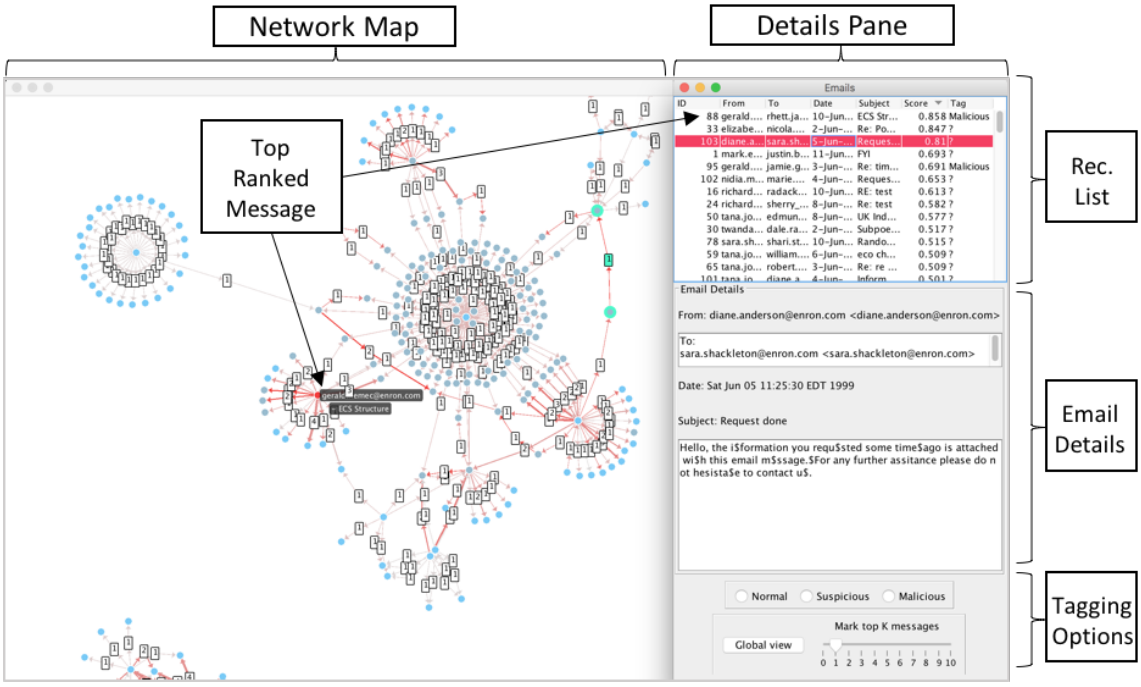


Figure 2. The experimental platform developed for the present study. The network map is in the left pane while the details pane is on the right. Note that the Top-K slider (bottom-right) is currently set to 1, which reveals the top ranked message on the network map. Also note "Rec." refers to "Recommendation."

We used a machine learning text classifier to determine the score for every communication in the network. This classifier was trained to detect anomalous content in the networks used in our experiments. Over the course of each trial it then learned to increase the scores of emails similar to those tagged as malicious and decrease the scores of emails similar to those tagged as normal. Scores were represented in two ways. First, a recommendation list was presented beside the network map that showed the score. Second, the score was represented by the degree of redness of the edges representing the emails, with more redness indicating a higher score. Scores were updated after each tag applied by the user.

A logging system recorded all click actions, as well as what object was clicked on. In addition to the action itself, the state of the viewport (i.e., the portion of the network map currently visible in the left panel of Figure 2) at the time of the action was also recorded. This included information such as the visible nodes, the percent of the network that was in view, and the centre position of the viewport within the network. From these details we were able to derive all zooming and panning behaviour for later statistical analyses.

A population of qualified security or network operations centre operators was not available to us. In their place we recruited a sample of engineering students as participants. The skills and competencies of these participants imposed a significant limitation on the complexity of both the simulated network and the experimental tasks. We scaled down the complexity of both the network and the tasks by using a small, static network, and by asking participants to perform a relatively simple search and inspection task. This was aimed at emulating the cognitive demands of expert operators in full-scale systems, which hinges on the assumption that the cognitive tolerance of novice operators is a fraction of that of experts. A formal analysis of the accuracy of this scaling procedure was not conducted, and therefore this represents a limitation in our study.

In both experiments, participants attempted to uncover a chain of malicious emails sharing common characteristics that were indicative of their maliciousness. These chains were island-hopping in nature, meaning that the target email jumped between adjacent users. This encouraged interaction first with the recommendation list to identify the first email in a chain, and then with the network map to explore adjacent areas of the network.

All research herein complied with the American Psychological Association Code of Ethics and was approved by the Institutional Review Board at the University of Toronto. Informed consent was obtained from each participant.

2.4.2. Passive Data Monitoring

We employed a *passive data monitoring* (PDM) approach with logged interaction data to form inferences regarding attentional switching. PDM derives data from interactions that are inherent to the task so that subsequent analyses can determine if patterns in user interaction align with certain cognitive events or states. PDM is an alternative to identifying cognitive events by actively collecting biometric data through methods such as eye tracking or electroencephalogram (EEG; Palmius et al., 2016). For example, engagement with novel stimuli has been found neurologically to be represented in the dorsal stream (i.e., the pathway in the visual cortex that directs motor actions; Janczyk & Kunde, 2010). Therefore, one could outfit participants with EEG and infer engagement periods by detecting when the subject was experiencing these neurological processes. However, the prospect of outfitting civilian network operators who work long, sedentary shifts with intrusive EEG equipment is unrealistic. The efficacy of PDM has been demonstrated in map navigation (Mac Aoidh, Bertolotto, & Wilson, 2012) and depression onset detection (Palmius et al., 2016).

PDM pairs well with the ANM domain for three reasons. First, integrating a robust logging program to passively monitor user interactions is relatively easy since the vast majority of the user's interactions are done at the desktop (Goodall, Lutters, & Komlodi, 2009). Second, information about the system state and the content of displays is also readily available and time-stamped (Corchado & Herrero, 2011). Finally, ANM is a highly interactive domain (Werlinger et al., 2010), allowing for sufficient operator behaviour to be recorded to make reliable inferences.

2.5. Experiment 1.1

2.5.1. Motivation

The principal objective of our first experiment was to examine the behavioural impact of attentional switches in ANM. Our secondary objective was to evaluate the efficacy of PDM in ANM.

2.5.2. Methods

2.5.2.1. Participants

We recruited eighteen engineering students via an emailed advertisement (9 Male, 9 Female, m_{age} = 21.5, *SD* = 2.89). None of the participants had prior experience in cyber-security or prior knowledge of the experimental platform, paradigm, or hypotheses. Participants were paid CAD 30.00 for two hours of participation.

2.5.2.2. Experimental Platform

The version of the platform used in Experiment 1.1 is the same as that presented in Figure 2 with a slightly lower contrast colour scheme. Participants had the option to sort the recommendation list by any of its columns. Users were able to use the Top-K message slider to provide spatial context for the emails that had the highest scores associated with them, which was useful for determining if there were any clusters of recommendations in the network map.

Users were able to explore the layout of the network by *zooming* and *panning*. Zooming allowed users to expand the network such that a smaller portion of it occupied a larger portion of the viewport (i.e., zooming-in). Panning allowed the users to remain zoomed-in and to click and drag the network map to bring adjacent areas into their viewport.

2.5.2.3. Scenario Design

Each scenario was constructed around one of four *worm attacks*. Worms are self-replicating codes that propagate through adjacent machines (P. Li, Salour, & Su, 2008). They represent a fairly intuitive class of cyber-attack that novices could grasp given sufficient training. Each worm was marked by a unique characteristic that reflected a real-world scenario. The first worm was marked by some of the text within the body of the email being replaced with punctuation marks. The second was marked by the text inviting users to click on a suspicious website URL. The third asked users to update their passwords for a financial server. The fourth worm had some text that was out of order. While the marker was consistent within each worm, the actual text and subject lines differed such that the exact same email was not simply being forwarded along. This forced the users to read the content of the emails rather than identify their structure at a glance. Users were tasked with flagging all of the emails in the chain as malicious.

2.5.2.4. Experimental Design

Two machine learning algorithms for updating the scores of emails were evaluated. However, it was not hypothesized that they would have any effect on attentional switches. Our results confirmed this and therefore the alternative algorithms will not be described further here.

No experimental manipulations used in Experiment 1.1 were pertinent to the goal of understanding attentional switching in ANM. Instead, we divided all actions within trials into two labels: *viewport*-actions and *non-viewport*-actions. Viewport-actions included panning and zooming inputs. Non-viewport-actions included tagging, sorting, and inspecting actions. There are four possible sequences of these two action-classes, which align with the three phases of an attentional switch, plus a control condition (see Figure 3). Each of the action-classes represents an independent variable and were analyzed as follows:

- Control: The time it took participants to initiate non-viewport-actions that were preceded by non-viewport-actions.
- Disengagement: The time it took participants to initiate viewport-actions that were preceded by non-viewport-actions.
- Shifting: Consecutive viewport-actions were treated as one action, with the initiation time being the time of the first action in the sequence, and the completion time being the completion time of the last action in the sequence. This was done because multiple pans or zooms were typically required for the participants to achieve their desired viewport.
- Engagement: The time it took participants to initiate non-viewport-actions that were preceded by viewport-actions.

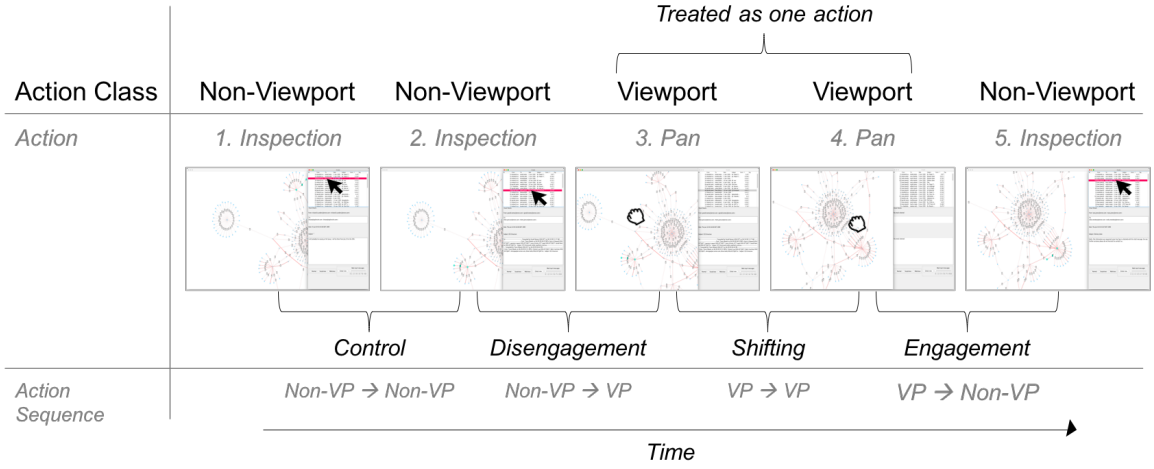


Figure 3. Timeline illustrating how the different sequences of action-class are treated in the data analysis. The figure illustrates how a viewport action (i.e., actions 3 and 4) changes the content of the viewport. The action latencies of interest are those of the second action in each of the action sequences (actions 2, 3, and 5).

The underlying rationale for using action latency as a measure of the impact of attentional switches is that if users experienced a cost from an attentional switch caused by either disengaging from their previous viewport or engaging with their new viewport, this should be reflected by an increase in action latency, during which the disengagement- or engagement-processes would be occurring.

We used completion time as the sole performance metric due to ceiling effects with tagging accuracy (average of 90% accuracy across participants and trials). Completion time was measured as the time of the final malicious tag in each trial. If participants failed to tag all emails in a worm, the completion time for that trial was set to the maximum trial length of 15 minutes.

Given our use of PDM over gathering biometric data (e.g., eye-tracking), our operational definition of an attentional switch deviates from that of Posner and Presti (1987), as we solely examined extra-viewport attentional switches. This operationalization sacrifices precision for practicality and accepts that we are missing attentional switches within viewports, asking the question of whether we can still characterize the effects of attentional switches in spite of this loss of precision.

2.5.2.5. Procedure

Participants were led through a PowerPoint presentation describing the components of the platform and their corresponding control actions. It also introduced them to the recommendation table and gave a high-level description of the machine learning algorithms driving the adaptations in the interface.

Participants were then introduced to the experimental task – searching for and tagging worms that consisted of between 5 and 8 emails hidden within the network. They were given examples of the four markers of worms that they would be looking for and told that each worm would be defined by one of these markers. Their principal objective was to find the origin of the worm and their secondary objective was to tag the rest of the emails in the worm as malicious. A strategy for how best to combine information in the recommendation table with the network map was described and demonstrated to participants. Participants were not trained to criterion. Instead, the experimenter judged their competency with the platform prior to advancing from the training phase. We do not feel that this represents a significant limitation in our research as our analyses focused on the execution of individual actions rather than on overall performance. Each participant completed four trials, and each trial had a single worm that the participant had not seen before.

2.5.3. Results

All actions with latencies of zero seconds were removed from analysis, as these represented double clicks. Following this, the top and bottom one percent of action latencies were removed from the dataset to remove both long pauses that were not representative of any engagement- or disengagement-processes and inadvertent clicks (Aguinis, Gottfredson, & Joo, 2013). This removal process retained actions with latencies between 0.015s and 10.08s.

Participants performed an action every 0.64s (SD = 1.27s) on average. The data were heavily skewed, so we used a Wilcoxon Rank-Sum test to identify differences in action latencies between the control actions and the actions corresponding to disengagements and engagements (see Figure 3). We found significant disengagement-delays (m = 1.45s; z = 28.60, $p < .0001$) and engagement-delays (m = 1.53s; z = 35.42, $p < .0001$) compared to control actions (m = 0.57s).

Figure 4 illustrates the distribution of action latencies across control actions, disengagements, and engagements.

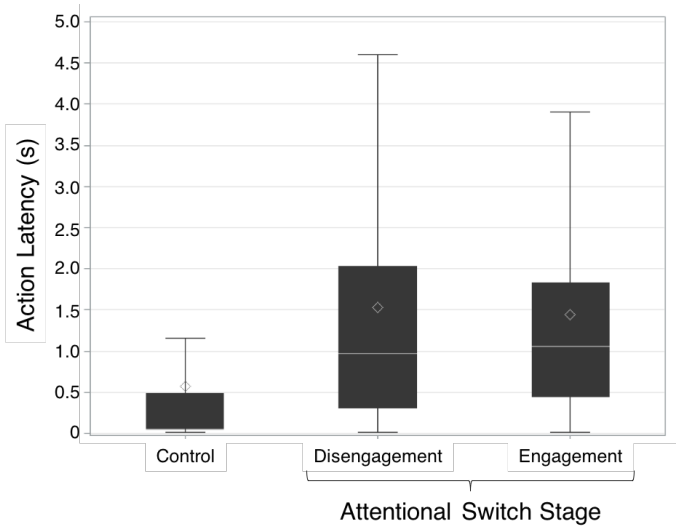


Figure 4. Boxplots showing the distribution of action latencies across control actions, disengagements, and engagements. All outliers have been removed from this graphic. Error bars show the maximum and minimum latencies following outlier removal.

We also compared average disengagement- and engagement-delays against completion time and did not find a significant relationship for either ($F(1, 16) = 1.23$, $p > .05$; $F(1, 16) = 2.75$, $p > .05$), indicating that the best and worst performers were equally susceptible to costs resulting from attentional switches.

2.5.4. Discussion

The results from Experiment 1.1 revealed both a significant disengagement-delay preceding and a significant engagement-delay following viewport-actions. This increase is large, nearly tripling action latency for both disengagement and engagement. This provides strong evidence that at a micro level (i.e., individual actions), viewport movement hinders performance and that limitations in human attentional switching capacity represent a cognitive bottleneck in ANM.

We did not find a significant relationship between either disengagement- or engagement-delay and performance. While this does not eliminate the limitation imposed by our novice participant pool, it does show that these delays are not artifacts of skill. Future studies should compare the effects of attentional switches between experts and novices.

Beyond identifying the micro performance impacts of viewport movements, Experiment 1.1 provided evidence for the effectiveness of PDM in ANM by successfully identifying the presence of a fundamentally cognitive phenomenon through behavioural indices. We believe that these methods can be extended to real-time monitoring of operator cognitive states, which can trigger adaptations to the user interface.

2.6. Experiment 2.2

2.6.1. Motivation

Our first experiment demonstrated that we could characterize micro performance impacts of attentional switches in ANM through PDM. Our second experiment builds on these results by assessing the cumulative impact of attentional switches in ANM and by seeking to facilitate attentional switches by improving the *visual momentum* in the interface. Visual momentum is the ease of extracting and integrating information when operators move to a new point in a display (Bennett & Flach, 2012; Woods, 1984). Moving to a new location requires operators to disengage from their previous location, shift to the new location, and then to engage with that new location. This process aligns with the three phases of an attentional switch described by Posner and Presti (1987), which suggests that attentional switches may be influenced by visual momentum. Woods (1984) posited that a key to increasing visual momentum is to provide the user with context that will allow them to more easily discern where they are in the interface relative to their previous location. We therefore implemented interventions that were aimed at providing this context to users when they moved to a new point in a display under the hypothesis that this would lessen the disengagement- and engagement-delays resulting from an attentional switch.

2.6.2. Methods

2.6.2.1. Participants

Nineteen participants (11 Male, 8 Female, m_{age} = 23.16, SD = 3.02) were recruited via an emailed advertisement. The participants were engineering students with no prior experience in cyber-security or prior knowledge of the experimental platform or paradigm. None of the participants were involved with the first experiment. Participants were paid CAD 40.00 for two hours of participation.

2.6.2.2. Experimental Platform

The main difference between the experimental platform used in our second experiment and the one used in our first was that we added a minimap, which allowed users to see where their current viewport fell within the larger network (see Figure 5). The minimap could also show how a tag impacted scores throughout the entire network by showing widespread colour changes beyond the user's current viewport. We built navigation control into the minimap so that the users could effectively *jump* to new points in the network by clicking on the corresponding point in the minimap, allowing for more efficient network navigation.

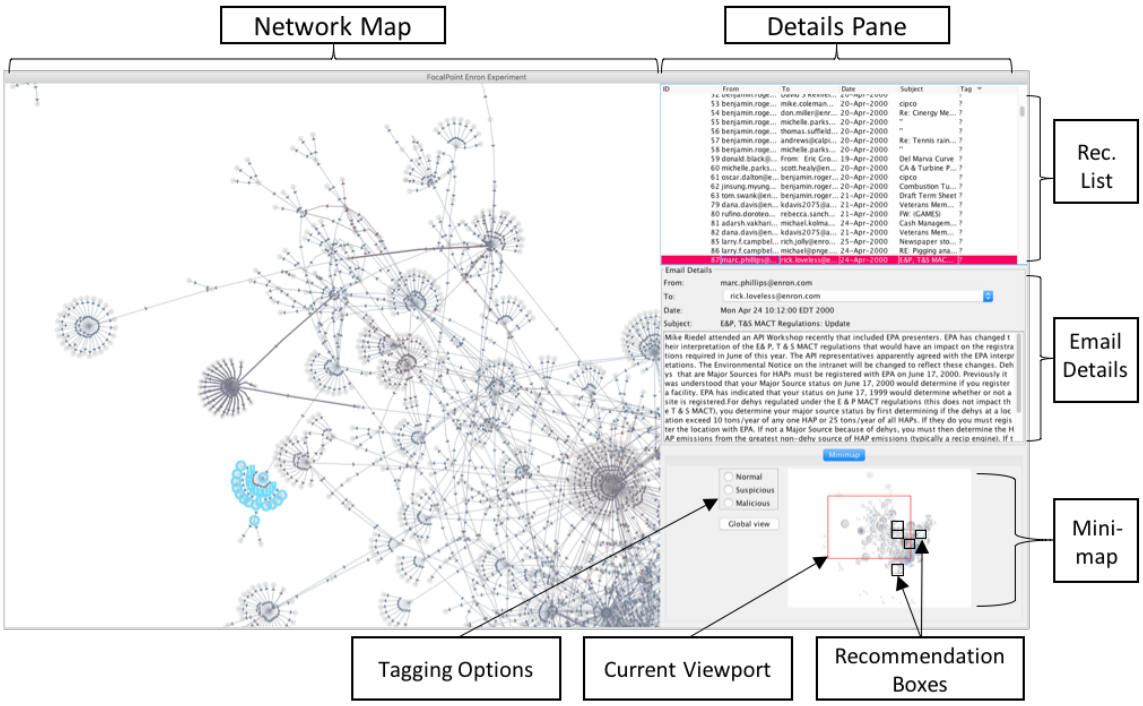


Figure 5. The experimental platform used for Experiment 2.2. The main changes that are visible are the size of the network and the presence of the minimap. Note that the recommendation boxes have been darkened for the purpose of this figure.

We overlaid recommendation boxes on the minimap as a new method for displaying recommendations. These boxes suggested areas for inspection rather than individual emails. The recommendation boxes were a function of the average score of the emails within each possible box, but with a minimum size such that the system didn't simply recommend individual emails with high scores. Users could click on the recommendation box and their viewport would shift to

that area. These were added to provide the users with additional capacity to use the minimap for navigation.

Our efforts to improve the visual momentum in the interface centred on the degree of context that was provided to the user when they navigated to a new area of the network via the minimap. This context was addressed through two interventions. First, we tried to increase the similarity between successive viewports when a user navigated via the minimap by assigning a proximity coefficient to the recommendation boxes such that they clustered around the user's current viewport. Second, we tried to provide users with a better understanding of the directionality and distance of their minimap navigations by implementing a sweeping transition wherein the screen would smoothly pan to a new location in a continuous motion. These two interventions were respectively contrasted by conditions where the recommendation boxes had no proximity coefficient and where the screen simply updated in a discontinuous cut when users navigated to a new location in the interface via the minimap. In each user trial the interface had one of the two recommendation methods (proximity or global), and one of the two transitional methods (sweep or snap). These are summarized below:

- Recommendation Method:
    - Proximity conditions: Recommendation boxes are the product of both the malicious content in the box and the proximity of the box to the user's current viewport.
    - Global conditions: Recommendation boxes are only the product of the malicious content in the box.
- Transitional Method:
    - Sweep conditions: Continuous panning transitions when navigating to new areas of the network via the minimap.
    - Snap conditions: Discontinuous cut when navigating to new areas of the network via the minimap.

In addition to adding features associated with the minimap, we also increased the size of the network from 150 to 500 emails. This decreased the signal to noise ratio to better emulate real-world operating scenarios and encouraged the use of the minimap. We also removed the score column from the recommendation table and the Top-K slider. These decisions were based on

pilot testing and sought to encourage reliance on the network map, which would be consistent with how network operators are observed to behave (Werlinger et al., 2010). A full description of the experimental platform used in Experiment 2.2 is provided in Kortschot et al. (2017).

2.6.2.3. Scenario Design

We modified the scenarios from our first experiment in order to force participants to read the content of the email rather than simply scanning it for some of the characteristics that they had been warned about. In the modified scenarios participants sought to identify a chain of users discussing a suspicious subject. The nodes representing the users who were involved in the target conversations were distributed over a greater portion of the network compared to the first experiment, requiring more navigation. Each scenario retained the underlying island-hopping structure of the first experiment, but had *linearly connected* emails, meaning that users who received an email in the conversation never responded to the person who sent them that email, and instead sent a new email to a new member of the chain.

Each of the four scenarios involved users talking about something that was either illegal or against company protocol. The two illegal scenarios involved users discussing an embezzlement scheme or discussing leaking confidential information to the press, respectively. The two protocol violation scenarios involved users requesting and sharing passwords through email or discussing confidential information (with no mention of leaking it to the press), respectively.

2.6.2.4. Experimental Design

The two intervention dimensions (i.e., recommendation and transitional method) resulted in a 2 X 2 within-subjects experimental design wherein participants completed four trials, each with a different pairing of recommendation and transitional method, and with a single scenario that they had not encountered before. All trials and conditions were randomized and counterbalanced to account for any learning effects.

2.6.2.5. Measures

Both interventions were centred on use of the minimap. As such, our analyses focused on minimap-navigations rather than viewport-navigations. The breakdown of action-classes was identical to Experiment 1.1 in that we examined disengagement- and engagement-delays on

either side of a shift (see Figure 3). However, instead of the *shift* action being panning and zooming, we focused on *jumping* via the minimap. Using this method we observed how the abovementioned interventions influenced the different phases of an attentional switch. Control actions remained identical to Experiment 1.1.

In addition to examining the effects of individual attentional switches, we also sought to characterize the cumulative effect of these switches on task performance. To do this, we examined the average action latency, the total viewport movement, and the number of switches– all across an entire trial, and compared them to the completion time of that trial. Completion time was measured as the time of the last malicious tag given by the user and was used as the principal performance metric. Once again, we observed ceiling effects with accuracy, with only 25 false tags out of the 380 total tags over the experiment.

2.6.2.6. Procedure

The procedure of Experiment 2.2 was similar to Experiment 1.1 with some key differences. Training was delivered through a narrated video of a modified version of the PowerPoint from the first experiment. Participants were encouraged to pause the video and ask questions of the experimenter, who also demonstrated some of the concepts on a sample network during planned pauses. The sole experimenter judged when participants were ready for experimentation. We do not believe that this presents a significant limitation as we were again focused on micro-behavioural measures related to switch cost rather than examining experimental performance alone.

Prior to each trial the participant was alerted to what marker they would be searching for. This was done after pilot testing revealed that participants had substantial difficulty finding the target conversations, which looked very similar to benign emails at first glance. Following each trial, participants completed a NASA-TLX workload rating scale (Hart & Staveland, 1988) and a system usability scale (SUS; Brooke, 1996). These were included to assess the relative differences in workload or usability resulting from the experimental manipulations and not to compare against industry standards.

2.6.2.7. Results

We used the same outlier removal procedures from Experiment 1.1 to eliminate long pauses that were not reflective of disengagement- or engagement-processes, as well as double clicks. Again, a Wilcoxon rank-sum test was used for several of our analyses due to skewed data and an inability to fit mixed models to that data. Relative to the control condition (m = 0.38s, SD = 0.93s), we found significant disengagement-delays (m = 2.68s, SD = 1.51s; z = 30.91, $p < .0001$), and significant engagement-delays (m = 2.60s, SD = 1.51s; z = 33.24, $p < .0001$) resulting from attentional switches via minimap navigation. Figure 6 illustrates these results.

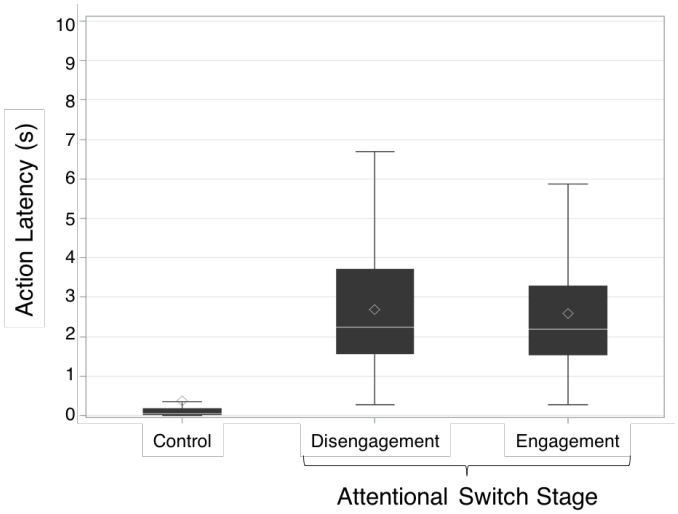


Figure 6. Boxplots showing the distribution of action latencies across control actions, disengagements, and engagements. All outliers have been removed from this graphic.

We did not find a significant relationship between disengagement- and engagement-delays (F(1, 447) = 0.30, $p > .05$), indicating that pausing longer prior to making a jump did not alleviate the engagement-delay following that jump.

Generalized linear mixed models on disengagement- and engagement-delays between experimental conditions found a significant effect of transitional method on disengagement-delay, with sweeping transitions (m = 2.22s, SE = 0.03s) shortening disengagement-delays relative to snapping transitions (m = 2.60s, SE = 0.03s; F(1, 374) = 8.25, $p < .001$). The sweeping transitions also had a significant reduction in engagement-delays (m = 2.21s, SE = 0.03s) compared to the snap conditions (m = 2.46s, SE = 0.03s; F(1, 442) = 4.69, $p < .05$). Figure 7

illustrates these results. We did not find an effect of recommendation method on disengagement-delays ($F(1, 374) = 0.01$, $p > .05$) or engagement-delays ($F(1, 442) = 0.03$, $p > .05$).

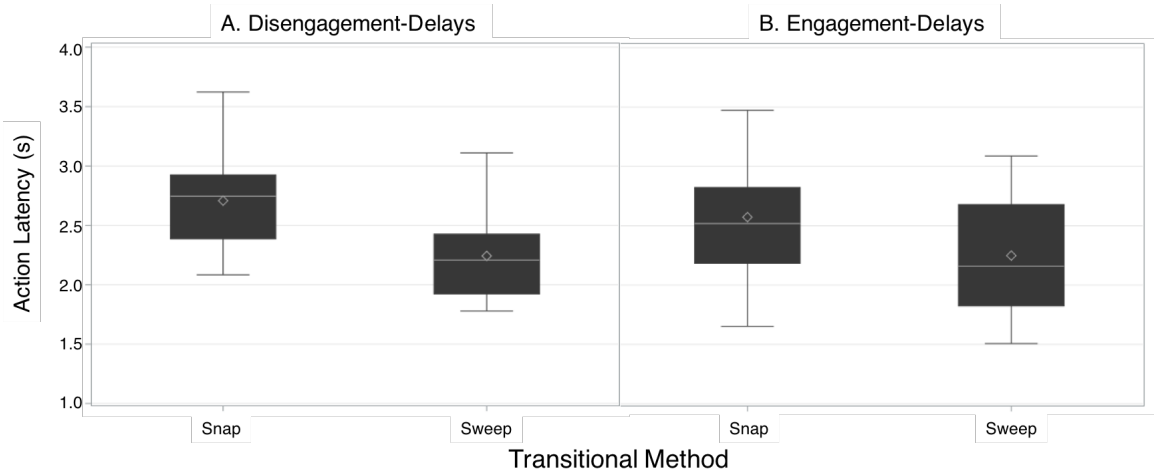


Figure 7. A. Boxplots showing the differences in disengagement-delays between the two transitional methods used in Experiment 2.2. B. Boxplots showing the differences in engagement-delays between the two transitional methods used in Experiment 2.2. The transitional method was significant in both A. and B. All outliers have been removed from this graphic.

Having again observed micro-performance decrements of attentional switching, we then examined the cumulative, macro-performance impacts. To do this, we ran mixed linear models examining the relationship between completion time, average action latency, total viewport movement, and total number of actions. Completion time was measured as the time of the fifth and final malicious tag in each trial. If participants failed to tag all five emails the completion time was set to the maximum trial length of 15 minutes. A trial's average action latency was the average amount of time between successive actions across a trial. The total movement within a trial was the summation of both zooming and panning behaviour, rescaled between 0 and 1 since zooming and panning were measured on different scales.

There was a significant negative relationship between the total number of actions and both the average action latency within trials ($F(1, 48) = 26.54$, $p < .0001$) and the total movement within a trial ($F(1, 48) = 7.59$, $p < .01$). There was also a significant positive relationship between total number of actions and completion time ($F(1, 48) = 55.12$, $p < .0001$). The average action latency across a trial had a positive relationship with the total viewport movement within that trial ($F(1, 48) = 10.69$, $p < .01$) but not with the completion time of that trial ($F(1, 48) = .02$ $p > .05$). We

did not find a significant relationship between the total viewport movement in a given trial and the completion time of that trial ($F(1, 48) = 1.02$, $p > .05$). Figure 8 shows the correlation matrix between the abovementioned variables.

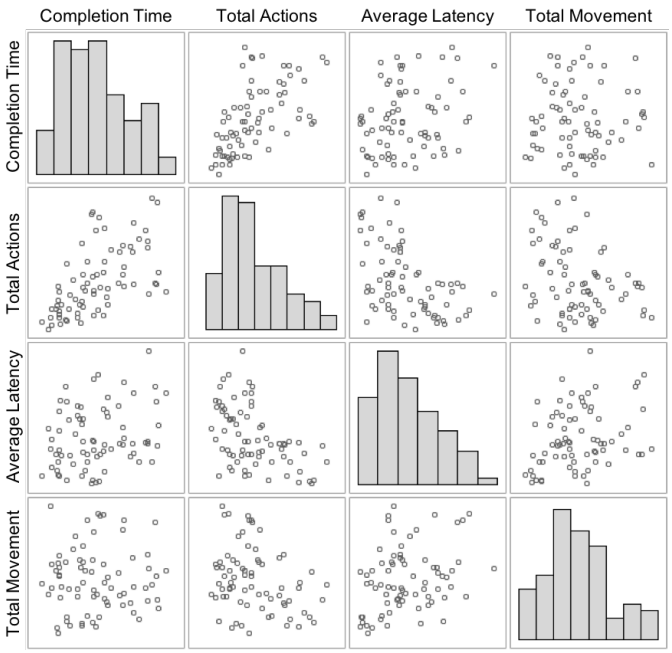


Figure 8. A correlation matrix showing the relationships and distributions between completion time, total number of actions, average action latency, and total movement. Each point represents a single trial.

There was no effect of either the recommendation method ($F(1, 46) = 0.43$, $p > .05$) or the transitional method ($F(1, 46) = 0.15$, $p > .05$) on completion times. However, there was a significant interaction effect between recommendation method and transitional method ($F(1, 46) = 7.15$, $p = .01$), indicating that the proximity recommendation method yielded faster completion times when using snap transitions but slower times when using sweeping transitions.

We also found that the transitional method significantly impacted the average action latency across a trial ($F(1, 46) = 7.49$, $p < .01$), with snap conditions eliciting shorter latencies on average ($m = 0.41$s , $SD = 0.04$s) than sweep conditions ($m = 0.47$s, $SD = 0.04$s). We did not find that either the recommendation method or the transitional method had any impact on the total movement within a trial ($F(1, 46) = 0.07$, $p > .05$; $F(1, 46) = 1.99$, $p > .05$).

We computed a single overall workload measure for each trial (Byers, Bittner, & Hill, 1989) and found that neither transitional method ($F(1, 48) = .20$, $p > .05$) nor recommendation method ($F(1, 48) = 2.31$, $p > .05$) were significant predictors of workload. However, we did find a significant interaction effect ($F(1, 48) = 5.00$, $p < .05$), indicating that sweeping transitions yielded lower workload ratings in the global recommendation method but higher ratings in the proximity recommendation method. There were no significant impacts on SUS scores across transitional method ($F(1, 48) = .06$, $p > .05$), recommendation method ($F(1, 48) = .04$, $p > .05$), or for the interaction between the two ($F(1, 48) = 3.02$, $p > .05$).

2.6.3. Discussion

Our results replicated the findings from Experiment 1.1, showing that participants succumbed to both disengagement- and engagement-delays when moving to a new viewport. These delays were longer in Experiment 2.2, likely as a result of the added capacity to make larger jumps.

The panning transition was designed to increase visual momentum by allowing users to grasp where they have navigated to relative to where they previously were (Woods, 1984). In the snap conditions, it may have been difficult for the operators to grasp any directionality of the transition, and it therefore took additional time to engage with the new viewport following a jump. We also found that the sweeping conditions facilitated disengagement processes by showing a significant reduction in disengagement-delays relative to the snap conditions. We believe this result stems from improved visual momentum reducing the degree to which participants needed to prepare for switches. We did not find an effect of recommendation method on engagement-delays, contradicting the transitional method finding. We had anticipated that disengagement- and engagement-delays would be lessened in the proximity recommendation conditions due to new viewports being closer to prior ones, but this was not borne out by evidence. This suggests that in spite of the shared context, a screen update requires significant disengagement and engagement irrespective of the distance between successive screens.

Our second experiment also characterized the cumulative effect of attentional switches across an entire trial. We found a significant negative relationship between action count and action latency, indicating that as people sped up their actions, they performed more of them. Interestingly, we did not find a significant relationship between average action latency and completion time, which shows that as people sped up their actions, they did not complete trials faster. These findings

show that improving the speed at which a user interacts with the system on an action-by-action basis does not necessarily improve the speed that they complete a task. Therefore, the wastefulness of actions needs to be considered when attempting to improve operator temporal performance.

The significant relationship between the total viewport movement in a trial and the average action latency from that trial indicates that the more people move throughout a network, the slower their actions become. It is consistent with our micro findings to attribute this to the cumulative effect of disengagements and engagements across a trial. Surprisingly, we did not find a relationship between total movement and completion time or between average action latency and completion time. However, based on the strong correlation between action latency and movement, we suspect that this relationship does exist. This hypothesis needs to be studied further to make concrete conclusions and design recommendations.

We found a significant interaction effect between recommendation method and transitional method on completion time, showing that in the sweep conditions, the global recommendation method yielded the fastest completion times whereas in the snap conditions, the proximity method yielded faster completion times. We believe the most likely explanation for this effect is that in the proximity conditions the minimap navigations were to closer areas, as the recommendation boxes were clustered around the user's current viewport. Participants may have therefore needed less context when navigating to those proximal areas, thereby reducing the need for a contextualizing transition. We also found that across a trial, the snap conditions elicited shorter average action latencies than the sweep conditions, indicating that the snap conditions accelerated the rate at which users interacted with the system. It is unclear why this effect occurred and should therefore be studied at greater depth in future experiments.

Workload was not significantly impacted by either transitional method or recommendation method. We did find a significant interaction suggesting that in the global recommendation conditions, sweeping yielded lower workload scores whereas in the proximity recommendation conditions, snapping yielded lower scores. Again, we believe that this is the result of the recommendation boxes being further away in the global recommendation conditions, and therefore the sweeping transition lowering the cognitive load of navigating with the minimap. In

the proximity recommendation conditions, where minimap navigations were closer, the sweeping transition was less necessary.

We did not find a significant effect of either transitional method or recommendation method on system usability. This suggests that participants were tolerant to the various interventions. This non-significant result is telling. As participants did not find the interventions (i.e., sweeping, proximity) to be less usable compared to the controls (i.e., snapping, global) in spite of their relative novelty, we believe that new interventions aimed at lessening the cognitive load of operators should be sought after, regardless of their potential novelty to users.

2.7. General Discussion

Taken together, the results of our two studies demonstrate the adverse effects of attentional switching in ANM at both the micro and macro levels. We provide evidence that the findings from the laboratory studies summarized in much of the attentional switching literature generalize to a more complex task environment. We also showed that the disengagement- and engagement-phases of an attentional switch outlined by Posner and Presti (1987) translate to navigational tasks. Our results suggest that attentional switches may be prevalent in real-world tasks and that the cumulative effects of these switches could be detrimental to performance.

It should be noted that trials in our study were limited to 15 minutes. The net effects of attentional switches are likely greater when they accrue across a full work shift. This suggests that attentional switches represent an important aspect of operator performance and should therefore be considered in the design of future DSSs. Future work should examine the costs of attentional switches associated with real-world, non-navigational tasks such as opening a new window or switching to a different application over the course of full operator shifts.

An interesting finding from our experiments was the self-imposed preparation (i.e., disengagement) that participants demonstrated. Previous literature has demonstrated that increasing preparation time facilitates attentional switching (e.g., Sohn & Anderson, 2001), but rarely do these experiments allow for participants to assume these preparatory periods voluntarily. Recently, Longman et al. (2017) showed that voluntary preparations reduced the costs of a switch. We did not find a significant relationship between the duration of disengagement-delays and their corresponding engagement-delays. However, the fact that

participants were consistently incurring such a significant disengagement-delay suggests that this delay either serves a purpose or is the result of a limitation in human cognition. Future studies should examine whether this behaviour is exhibited across other attentional switches and whether or not this relationship can be facilitated through design.

The studies reported here also demonstrate the practicality and effectiveness of PDM within ANM. The cost of attentional switching has been difficult to study in real-world scenarios largely due to the need for highly sensitive time measures. However, if the user's interactions are easily logged, PDM can overcome this issue. Beyond PDM's ability to generalize time-sensitive measures like attentional switch detection to real-world domains, we believe that it represents a promising method for inferring cognitive states. There has been a recent trend in the literature towards biometric readings for cognitive measurement, largely motivated by the improved access and accuracy of these measures (Verwey, Shea, & Wright, 2015). Although increasingly accessible, biometric readings are still far more invasive than PDM (Balakrishnan, Durand, & Guttag, 2013). PDM can be implemented on many systems and imposes virtually no additional load on the operator. It is therefore highly feasible for domains where operators work long, sedentary shifts and wear minimal equipment. Future studies should continue to develop methods of inferring cognitive phenomena and states through the use of logged operator interaction data.

While the current study demonstrated the efficacy of PDM for inferring cognitive events (i.e., disengagements and engagements), future studies should look at inferring cognitive states (e.g., stress, boredom, etc.). These are less likely to have direct behavioural manifestations and clear onsets and offsets and would therefore demand machine learning algorithms to be run on more continuous data streams such as cursor position. However, in spite of the increased difficulty, the potential utility of this research is extensive and can be directly applied to triggering adaptive interface behaviours based on operator measurement through PDM (Feigh et al., 2012).

2.8. Conclusion

As machine learning and reasoning become more prevalent in ANM, the psychological demands of the human operators need to be considered to optimize the joint human-machine task performance. We have shown here that these demands can be successfully incorporated into the interfaces and algorithms that drive the DSS through the use of PDM.

Our study focused solely on attentional switching within ANM. However, with the growing size and complexity of both computer networks and the attacks on those networks, the psychological demands imposed on the operator will continue to grow. Therefore, in order to realize the full potential of the machine learning driving DSSs, a full spectrum of cognitive states should be studied and incorporated into the design of future systems.

Chapter 3 - Attentional Tunneling

The first phase of this dissertation demonstrated that an attentional event can be measured through PDM and accounted for in the design of a user interface. Phase 2 sought to build off of these findings by evaluating whether an enduring attentional state can also be detected through PDM. Phase 2 sought to detect *attentional tunneling*, which is "…the allocation of attention to a particular channel of information, diagnostic hypothesis, or task goal, for a duration that is longer than optimal, given the expected cost of neglecting events on other channels, failing to consider other hypotheses, or failing to perform other tasks" (Wickens & Alexander, 2009; p. 182). Attentional tunnelling was expected to manifest behaviourally in more obvious patterns than the Phase 3 attentional state, information overload, as we expected participants' interactions to be more concentrated when attentionally tunnelled compared to a baseline state. Therefore, attentional tunnelling was intended to serve as a useful transition from an attentional event with a clear onset and offset (i.e., attentional switching) to an attentional state with less predictable behavioural manifestations (i.e., information overload).

Phase 2 did not employ an intervention experiment (see Figure 1). This omission was motivated by the overarching goal of our industry partner Uncharted Software, which was to develop Adaptive Level of Detail (ALOD) displays. ALOD displays autonomously adapt either the amount or degree of aggregation of information that is presented to users. Because an attentional tunneling intervention was unlikely to incorporate an ALOD design, and because Phase 3 had already been planned and would incorporate this design, Phase 2 was limited to developing machine learning methods that could be used with PDM. (A Masters student will complete the intervention experiment.)

3.1. Statement of Authorship

This chapter has been accepted for publication in Human Factors (Kortschot & Jamieson, in press) and was authored by Sean W. Kortschot and Greg A. Jamieson. Under the supervision of Greg A. Jamieson, Sean W. Kortschot conceptualized the research idea, defined the scope of the research, conducted the literature review, created the research design, and built the testbed used for experimentation. Sean W. Kortschot conducted all experimentation through Mechanical Turk and selected, developed, and interpreted all statistical and machine learning models. Sean W.

Kortschot wrote and revised the majority of this paper with Greg A. Jamieson providing editorial oversight throughout the writing and revision processes.

Classification of Attentional Tunneling Through Behavioural Indices

3.2. Abstract

Objective: Develop a machine learning classifier to infer attentional tunneling through behavioural indices. This research serves as a proof of concept for a method for inferring operator state to trigger adaptations to user interfaces.

Background: Adaptive user interfaces adapt their information content or configuration to changes in operating context. Operator attentional states represent a promising class of triggers for these adaptations. Behavioural indices may be a viable alternative to physiological correlates for triggering interface adaptations based on attentional state.

Method: A visual search task sought to induce attentional tunneling in participants. We analyzed user interaction under Tunnel and Non-Tunnel conditions to determine if the paradigm was successful. We then examined the performance trade-offs stemming from attentional tunnels. Finally, we developed a machine learning classifier to identify patterns of interaction characteristics associated with attentional tunnels.

Results: The experimental paradigm successfully induced attentional tunnels. Attentional tunnels were shown to improve performance when information appeared within them, but to hinder performance when it appeared outside. Participants were found to be more tunneled in their second Tunnel trial relative to their first. Our classifier achieved a classification accuracy similar to comparable studies (AUC = 0.74).

Conclusion: Behavioural indices can be used to infer attentional tunneling. There is a performance trade-off from attentional tunneling, suggesting the opportunity for adaptive systems.

Application: This research applies to adaptive automation aimed at managing operator attention in information-dense work domains.

3.3. Introduction

Decision support systems (DSSs) have become commonplace in modern control rooms (van Dongen & van Maanen, 2013). The principle role of the DSS is to aid operators in effectively extracting meaningful insights from the often copious information that is available to them. DSSs thus serve as attentional guidance tools by highlighting important or anomalous behaviour in the information space of interest (Corchado & Herrero, 2011). In spite of advances to DSSs, the attentional load imposed on operators in many domains remains significant. This presents an opportunity for *adaptive user interfaces*, which adapt their information content, configuration, or interactions to facilitate operator performance and workload (Feigh et al., 2012).

Adaptations can be *triggered* by either spatio-temporal changes, or changes in the system, environment, task, or operator (Feigh et al., 2012). We focus here on the *operator measurement* class of triggers, which respond to an operator's mental or physical state. Feigh et al. (2012) define two methods for measuring an operator's state: Physiology and performance. Physiological state inference relies on technologies such as electroencephalograms (EEG) or electrocardiograms (ECG), an example being the *Communication Scheduler*, which uses EEG and ECG to infer soldiers' workload to modulate communications (Dorneich et al., 2005). Performance state inference relies on real time measures of operator performance assuming an inverse proportionality to workload (Feigh et al., 2012). A central challenge limiting the implementation of performance-based operator state inference, however, is that most task domains lack an accurate and continuous performance metric. This lack, coupled with increased accessibility and accuracy (McLane et al., 2015; Verwey et al., 2015) as well as decreasing invasiveness of apparatuses (Mukhopadhyay & Lay-Ekuakille, 2010), has led most research on operator state inference to focus on physiological measures. Some notable examples of physiological state inference include detection of mind wandering (Baldwin et al., 2017), task engagement (Berka et al., 2007), mental workload (Borghetti et al., 2017; Wilson & Russell, 2004), and various emotional states (Y. Y. Lee & Hsieh, 2014). While these studies demonstrate the promise of physiological state inference, practical applications still require initial investment and users willing to use the necessary recording devices and share their biometric data.

A third approach for operator measurement, not described in Feigh et al. (2012), is Passive Data Monitoring (PDM), which focuses on *behaviour* rather than *performance* by examining streams

of interaction data that are inherent to a task (Kortschot et al., 2018; Palmius et al., 2016). PDM exploits data that exists irrespective of performance (e.g., cell phone location from GPS, cursor position in desktop applications, etc.) and examines it for patterns that align with cognitive events or states. PDM is not intended to replace physiological inference in all domains. Instead, it represents a viable alternative in domains with behavioural data that is both rich and readily available and where outfitting operators with biometric recording devices is impractical, unfeasible, or unnecessary. PDM sacrifices some of the precision that biometrics can offer (e.g., cursor tracking is a less accurate measure of attentional focus than eye tracking) for significantly increased practicality. The efficacy of PDM for inferring cognitive events or states has been demonstrated for attentional switching (Kortschot et al., 2018), depression onset detection (Palmius et al., 2016), information overload (Mac Aoidh et al., 2012), and driver drowsiness detection (McDonald et al., 2014).

Both PDM and physiological state inference share the fundamental challenge of labelling what data belong to what cognitive state. Palmius et al. (2016) labelled impending depressive episodes via weekly questionnaires. Although this method is well suited to enduring cognitive states, it is incompatible with more transient states that adaptive automation typically targets. Early work in physiological state inference solved the labelling issue by inducing a state and examining how peoples' physiology responded (Kramer, 1991). A similar approach can be used for PDM.

The attentional state that we focus on in the present research is Attentional Tunneling, which Wickens and Alexander (2009) operationalized as "…the allocation of attention to a particular channel of information, diagnostic hypothesis, or task goal, for a duration that is longer than optimal, given the expected cost of neglecting events on other channels, failing to consider other hypotheses, or failing to perform other tasks" (p. 182). We operationalize attentional tunnelling slightly differently, describing it in relation to the *potential* cost of neglecting events rather than the *expected* cost, as in many instances attentional tunneling can be conflated with directed attention, which can serve as an adaptive mechanism (Dixon et al., 2013). Attentional tunnelling is believed to be a significant contributor to the accident at Three Mile Island (Rubenstein & Mason, 1979), most aviation accidents involving controlled flight into terrain (Shappell & Wiegmann, 2003), and many other incidents.

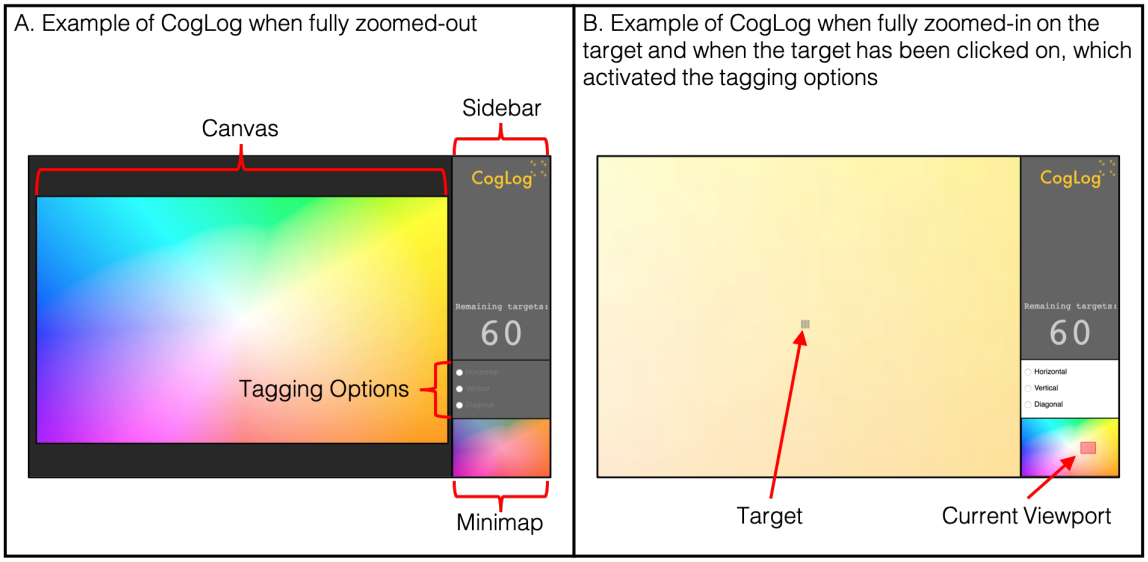
Several factors can precipitate attentional tunnels. These include increased cognitive load (Rantanen & Goldberg, 1999; Williams, 1995), display design (Wickens & Alexander, 2009), conversation (Atchley & Dressel, 2004), and operator fatigue (Mills, Spruill, Kanne, Parkman, & Zhang, 2001). Rogé, Kielbasa, and Muzet (2002) demonstrated that task priority and complexity can lead to attentional tunnels, with operators overly fixating on tasks that they perceive to be the most important. Many of these factors are prevalent in information dense spaces. Therefore, attentional tunnelling represents a promising candidate attentional state for treatment through DSS design.

This article presents an experiment aimed at inducing attentional tunneling and developing a machine learning classifier based on PDM that can detect when users are in that state. We examine the viability of this method in a low-fidelity environment and describe the performance trade-offs stemming from attentional tunnels. We hypothesize that the induction of an attentional tunnel will improve performance when information appears within that tunnel at the expense of degrading performance when information appears outside of it. Finding this performance trade-off will verify successful induction of an attentional tunnel and therefore validate that the resulting machine learning classifier is identifying attentional tunnels. This research serves as a proof of concept for PDM.

3.4. Methods

3.4.1. Experimental Platform

This study was conducted on a novel experimental platform called the Cognitive Logger (CogLog; Appendix B). CogLog is an online platform used to collect behavioural datasets for developing machine learning classifiers. The version of CogLog used in the present study (CogLog_AT.3) consisted of two main elements: a canvas and a sidebar (see Figure 9). The canvas was 1000 X 600 pixels and its background was set to a continuous colour wheel covering the full visible colour spectrum. Participants were able to *zoom in* on different areas of the canvas, which caused a smaller portion of the canvas to occupy a greater portion of the viewport (i.e., the portion of the canvas that was currently in view). They could also *pan across* the canvas by clicking and dragging, which caused areas of the canvas that were adjacent to the participants' viewport to be brought into view. By doing this, participants were able to search for the presence of targets that appeared, one at a time, in random locations on the canvas.



** Note that this image was originally in colour*

Figure 9. CogLog platform used for experimentation. All red brackets, arrows, and labels are shown for illustrative purposes only. Note that the target in Figure 9B has been darkened for the purpose of this figure. The three tagging options are Horizontal, Vertical, and Diagonal.

Targets were small (3 X 3 pixel) striped patches containing either horizontal, vertical, or diagonal lines. They had an opacity of 12% with a *multiply mix-blend-mode*, which caused their RGB values to be multiplied by the RGB values of the section of the canvas on which they appeared (Budd & Björklund, 2016). For example, if a target appeared in a red area of the canvas, the red values of the target would be increased disproportionately to the green and blue values, thereby causing the target to render with a red tint. This allowed for approximate equivalence of detection difficulty regardless of the background colour on which the target appeared. The size and transparency of targets ensured that participants would have to zoom in and scan subsections of the canvas for successful detection and classification.

The sidebar consisted of a counter, tagging options, and a minimap. The counter indicated how many targets remained until the end of the experiment. Clicking on a target activated the tagging options (See Figure 9B), which were radio buttons that allowed participants to classify the target as being horizontal, vertical, or diagonal. The minimap allowed participants to see where their current viewport fell within the overall canvas with a red, translucent square (See Figure 9B).

CogLog recorded all user interactions, which included cursor position relative to both the screen and canvas, the location and size of the canvas relative to the participant's screen, and the classifications made by the participant. From these data we were able to derive all zooming and panning behaviours. CogLog sampled user data every time they executed an action (i.e., moved their cursor or moved the canvas). This was done rather than sampling at fixed rates so that we could later transform the data to sample at any rate less than the maximum sampling rate of the logger.

CogLog is a *behavioural analog* to a cybersecurity microworld simulation (see Kortschot et al., 2017), meaning that the input features (i.e., scroll to zoom, click and drag to pan) are identical between CogLog and the simulation. Therefore, demonstrating that attentional states can be classified using the suite of interactions in this environment will suggest the viability of the approach in higher-fidelity environments.

3.4.2. Experimental Design

The experiment employed a single factor within-subjects design. Participants each completed four trials, two in the *Tunnel* condition and two in the *Non-Tunnel* condition. There were two classes of targets used in both conditions: *Prime* targets, which were all targets in each trial except for the last target, and a *Test* target, the final target in each trial. Targets appeared one at a time with the next target generated upon the classification of the current target. We labelled the area in which targets could appear the *active area*, the boundaries of which remained unmarked to participants. In the Tunnel condition, the active area was a subsection of the canvas approximately one tenth the size of the total canvas area. The active area was randomly positioned at the start of each trial and all Prime targets appeared randomly within this region. In the Non-Tunnel condition, the active area for Prime targets occupied the entire canvas. The Prime phase of each trial was intended to *prime* an attentional state in the participant (e.g., Friedman, Fishbach, Förster, & Werth, 2003). The Test target in both Tunnel and Non-Tunnel conditions could appear anywhere on the canvas, provided that it was at least 300 x 200 pixels away from the final Prime target. Figure 10 illustrates the experimental conditions.

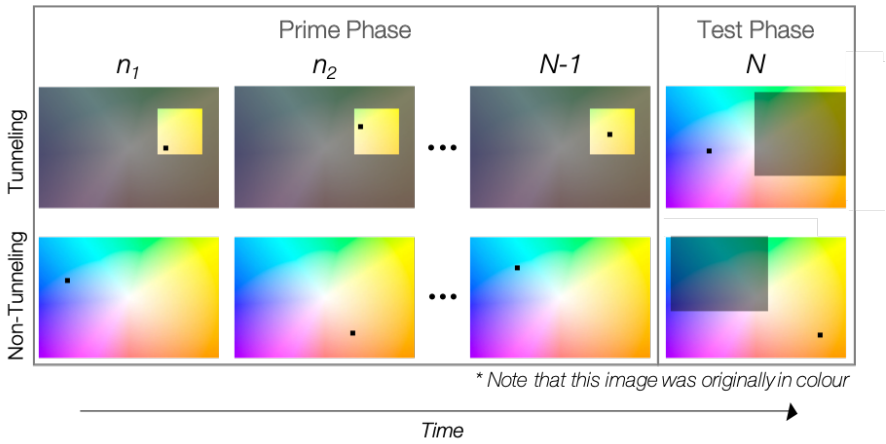


Figure 10. Top-row: The Tunneling condition. The active area (unshaded) was approximately 10% of the canvas size. All Prime targets appeared within this area. Bottom-row: The Non-Tunneling condition. The active area (unshaded) occupied the entire canvas for all Prime targets. In both conditions, the Test target could appear anywhere on the canvas outside of a 300 X 200 pixel box (shaded) from the final Prime target (N-1). For both conditions, N ranged from 13-16 at random. Targets here are darker and larger for illustrative purposes. The shaded area is also only shown for illustrative purposes.

Both the colour wheel that was used as the canvas background and the minimap were intended to implicitly remind participants where they were located in the canvas. Essentially, by having all Prime targets appear in a small box in the Tunnel condition, they also appeared in the same colour space. This was expected to bias participants to a particular region and colour such that navigating to a new area felt foreign during the Test phase.

During the Prime phase participants were never given any information about the distribution governing target generation. Therefore, if they behaved differently between conditions it could only be attributed to them updating their expectations about where future targets were likely to appear given where previous targets had appeared before. Furthermore, because there was no difference in how the Test target was generated between conditions, any difference in participant behaviour during the search for the Test target can solely be attributed to the distribution of the Prime targets that preceded it. Critically, this paradigm ensures that we are not distinguishing between behaviour from two different tasks, but rather between behaviour from the same task performed under two different attentional states.

The majority of the collected data came from the Prime phase of the study. This meant that there was insufficient data from the Test phase to develop a reliable machine learning classifier. We

therefore used traditional statistics to examine any differences in behaviour during the Test phase, which identified whether participants succumbed to attentional tunnels in the Tunnel condition relative to the Non-Tunnel condition. Our classifier was then trained on the Prime phase data, and sought to determine a pattern of behaviour that was indicative of a developing attentional tunnel.

3.4.3. Procedure

Experimentation was completed on Amazon's Mechanical Turk. Sixty participants were paid a flat rate of CAD 10.00, thereby incentivizing them to complete the task as fast as possible. Due to a database error, only data from 50 participants was used in our analysis. Following a description of the study's requirements, participants were given a screening questionnaire that presented sample targets on different areas of the canvas's colour-wheel. This was designed to test screen resolution, visual acuity, and colour vision and ensured that anyone who advanced to experimentation would be able to complete the task. Participants were then given an interactive training regimen that detailed how to interact with the interface, presented videos of an expert user performing the task, and presented participants with five practice targets to search for. Participants could only advance to experimentation once they had successfully classified the five practice targets.

Each participant completed four trials and classified a total of 58 targets. The number of targets within each trial ranged from 13-16 in random order and the sequencing of Tunnel and Non-Tunnel conditions was fully randomized. Although participants were never informed of the experimental condition that they were in and never saw the bounding box for the active area, astute participants may have inferred the two different distribution algorithms used to generate targets. They may have therefore realized that the final target in their previously experienced Tunnel trial fell outside of the Prime phase's active area. The counter (see Figure 9) therefore began at 60 rather than 58 so that it never informed participants they were on the last target of the experiment.

All research herein complied with the American Psychological Association Code of Ethics and was approved by the Institutional Review Board at the University of Toronto. Informed consent was obtained from each participant.

3.5. Results

Participants accurately classified 99.07% of possible targets. Therefore, all results will focus on the temporal aspects of performance rather than accuracy.

3.5.1. Data Cleaning

The data was transformed from the input-dependent method to a fixed sampling rate of one sample per 75 milliseconds. This forward filled stretches with no interactions and removed observations during stretches where the user was interacting at a rate higher than once every 75 milliseconds. For all statistical tests and for our machine learning classifier, we excluded all behaviour recorded during the search for the first target because the time it took for a participant to identify this target was an artifact of the participants' search relative to the random position of the target. We also excluded behaviour recorded during search for the second target because a tunnel could not have been reinforced after a participant only identified the location of one target. Since this reduced the number of targets in Tunnel trials, our performance metrics depict average search time per target rather than the completion time of the overall trials.

3.5.2. Characterization of Attentional Tunneling

In order to characterize whether the paradigm successfully induced attentional tunnels we needed to examine several aspects of behaviour during both Prime and Test phases. Unfortunately, no perfect measure exists for what behaviour represented an attentional tunnel. This is partially due to the remote nature of the experiment but also because of the relative novelty of studying attentional tunneling purely through behaviour. Therefore, we needed to look at several proxy measures and determine if, when taken together, they suggest an effect.

Our principal metric for characterizing attentional tunneling was the time that it took to classify the Test target. The hypothesis driving this analysis is that if participants developed an attentional tunnel during the Prime phase of Tunnel trials, they should expect the Test target to appear within that tunnel and therefore continue their search in that area, which should result in longer classification times for the Test target. The experimental paradigm likely induced different levels of attentional tunneling during the Prime phase depending on the trial and the participant. The degree to which an attentional tunnel was induced was expected to influence participants' expectations about where the Test target would appear. We therefore used

performance and *degree of interaction* during the Prime phase of each trial as proxy measures for the degree of attentional tunneling that was induced in that particular trial. We measured performance in the Prime phase of each trial as the amount of time that it took on average to classify a target within that trial. Degree of interaction was a measure of the net amount of cursor, pan, and zoom movement, all normalized between zero and one, per target during the Prime phase of each trial. We performed generalized linear mixed models specifying participant, trial number, Prime phase performance and degree of interaction as random variables. We found that it took participants significantly longer to find the Test target in the Tunnel condition (M = 16.48 seconds, SE = 0.09) compared to the Non-Tunnel condition (M = 13.36 seconds, SE = 0.10; $F(1, 148) = 4.55$, $p = 0.03$).

Our second measure for characterizing attentional tunnels was *Test Target Search Bias*, which is a measure of whether participants' search during the Test phase was biased towards one area of the canvas. We calculated this in Tunnel trials by taking the moment in the Test phase that no part of the preceding Prime phase's active area remained within the participants' viewports (see Figure 10). In Non-Tunnel trials, we calculated this by taking the moment in the Test phase that no part of that Test phase's inactive area remained within participants' viewports. Participants' Test Target Search Bias was significantly shorter in the Non-Tunnel condition (M = 5.54 seconds, SE = 0.09) compared to the Tunnel condition (M = 7.12 seconds, SE = 0.08; $F(1, 129) = 10.47$, $p = .002$).

To further test if the experimental design successfully induced attentional tunnels, we examined three additional proxies. First, we examined the average classification time per target between Tunnel and Non-Tunnel conditions to determine if participants updated their beliefs about future target generation. We found that it took participants less time on average to find an individual target in the Tunnel condition (M = 14.62 seconds, SE = 0.08) compared to the Non-Tunnel condition (M = 22.47 seconds, SE = 0.08; $F(1, 152) = 87.74$, $p < .0001$). Second, we examined the mean zoom level during search for the Test target to determine the average amount of the canvas that participants were searching at each point. We found a significant effect, with participants viewing a smaller portion of the canvas per logged action in the Tunnel condition (M = 18%, SE = 0.07) compared to the Non-Tunnel condition (M = 21%, SE = 0.07; $F(1, 135691) = 583.01$, $p < .0001$). Finally, we examined whether detection time of an individual target and degree of interaction within that trial's Prime phase were influenced by whether it was the first or

second Tunnel trial. We found significant effects for both, showing that participants found individual targets faster in the Prime phase of their second Tunnel trial (M = 10.72 seconds , SE = 0.07) while also interacting less (M = 0.58 degree of interaction, SE = 0.10) compared to their first (M = 13.63 seconds, SE = 0.07; M = 0.76 degree of interaction, SE = 0.11; $F(1, 49)$ = 10.72, p = .002; $F(1, 49)$ = 4.15, p = .047). However, there was no difference in the amount of time that it took for participants to find the Test target between their first (M = 20.54 seconds, SE = 0.09) and second trials (M = 19.54 seconds, SE = 0.09; $F(1, 48)$ = 0.01, p = 0.91). Interestingly, the opposite effects were found in Non-Tunnel trials, with participant showing no significant performance improvement between the Prime phase of their first (M = 21.11 seconds, SE =0.08) and their second trial (M = 19.25 seconds, SE = 0.08; $F(1, 49)$ = 2.86, p = 0.10), but a significant improvement in Test phase performance between their first (M = 21.19 seconds, SE = 0.09) and second trials (M = 17.32 seconds, SE =0.09; $F(1, 49)$ = 5.28, p = 0.026).

3.6. Discussion of Behavioural Findings

Taken together, the results from the first phase of our analysis indicate that the experimental paradigm successfully induced attentional tunnels in the Tunnel condition relative to the Non-Tunnel condition. Although none of the measures that we included are perfect representations of attentional tunneling, the fact that they nearly all point in the same direction and that they tend to align with the behavioural aspects of the Wickens and Alexander (2009) definition of attentional tunnels suggests that the intended effect was achieved.

We found that participants searched the area around the final Prime target significantly longer in the Tunnel condition, indicating that they were overly focused on the area where they believed the next target would most likely appear and that this focus persisted in the absence of new targets. This additional focus delayed their target detection time when searching for the Test target. This result, coupled with the finding that participants found targets within the Prime phase of the Tunnel condition significantly faster than in the Prime phase of the Non-Tunnel condition demonstrates a clear performance trade-off: Improved performance when information appeared within an attentional tunnel at the cost of poorer performance when information appears elsewhere. This supports our operationalization's focus on the potential rather than the expected cost of an attentional tunnel. The degree to which this trade-off is considered in the design of DSSs should be dependent on the domain of interest. For example, in a safety critical system like

a nuclear power plant, it may be advisable to deter attentional tunnels even if they improve performance the majority of the time as, in rare edge cases, they may allow dangerous operating conditions to develop.

Interestingly, participants exhibited more tunneled behaviour during the Prime phase of their second Tunnel trial compared to their first, suggesting a relationship between experience with a system and susceptibility to attentional tunnels. The stronger tunnels in the Prime phase of the second Tunnel trial resulted in improved performance, which did not carry over to the Test phase of that trial. This may suggest that the presumed performance gains were offset by stronger preceding tunnels. Furthermore, the finding that there was no significant performance improvement between participants' first and second Non-Tunnel trials suggests that participants only improved when there was a predictable algorithm responsible for generating targets. The degree to which these effects carry over to higher-fidelity simulations and over longer timespans where operators gain more experience with a system needs to be evaluated. However, the alignment of these results with the literature (e.g., Briggs, Hole, & Turner, 2018; Dixon et al., 2013; Wickens & Alexander, 2009) indicates that they are likely to translate.

3.7. Machine Learning Classification

The findings from our initial analyses suggest that the experimental paradigm was successful in inducing attentional tunnels in the Tunnel condition relative to the Non-Tunnel condition. We will now present a machine learning classifier trained on data from the Prime phase of the experimental paradigm. The goal of this classifier was to identify patterns of behaviour that were characteristic of an attentionally tunneled state relative to a non-tunneled state.

Unlike many other classification tasks (e.g., image classification), our paradigm does not assume perfect ground truth labelling. Although we determined that the experimental paradigm successfully induced attentional tunnels in the Tunnel condition overall, it was likely that there were some trials where it failed (see Li et al., 2015 for a related approach). Therefore, to ensure that the data that we were passing into the classifier was reflective of its label, we examined both the degree of interaction and the time required to identify a target within the Prime phase of both Tunnel and Non-Tunnel trials. These measures determined if participants updated their search strategy in Tunnel trials relative to Non-Tunnel trials. If they interacted with the platform more, or took longer per target in a Tunnel trial compared to their averages in Non-Tunnel trials, then it

was determined that trial failed to induce the desired attentional tunnel. We identified 13 such trials out of a total of 208 trials and did not include their data in our classifier.

3.7.1. Inference Techniques

Our data was multivariate time series data and consisted of the X/Y coordinates of the user's cursor relative to the canvas, the zoom level of the viewport, and the X/Y coordinates of the canvas itself. Using these measures we calculated 12 total features, which included the speed of the cursor and screen as well as binned features that essentially reduced the canvas to a 20 by 20 grid, thereby decreasing the resolution of the data to cancel out some noise. At no point was the location of the targets passed into the classifier. Therefore, all classification was based purely on how the participant interacted with the platform and hinged on participants updating their search behaviour as they received increasing evidence about the distribution of targets.

Long Short-Term Memory (LSTM) recurrent neural networks are well-suited to this type of classification task as they can represent past information in their internal states and use these states to process new information, but without suffering from the vanishing gradient problem that can limit traditional recurrent neural networks (Hochreiter & Schmidhuber, 1997; Lipton, Berkowitz, & Elkan, 2015). LSTMs control how internal states update and output at each time step via gating functions (Hochreiter & Schmidhuber, 1997), which allows them to maintain long term storage of internal states and therefore to exploit distant temporal dependencies within the data (Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2017; Prieto, Alonso-González, & Rodríguez, 2015; Zhao, Chen, Wu, Chen, & Liu, 2017). Furthermore, LSTM networks can be stacked under initial 1-Dimensional convolutional layers (CNN-LSTM) in order to improve their processing speed and their feature extraction capabilities (Karim, Majumdar, Darabi, & Harford, 2018; Pak, Kim, Ryu, Sok, & Pak, 2018; Tan et al., 2018).

3.7.2. Neural Network Architecture

We used an eight-layer CNN-LSTM to classify the dataset (see Table 1 for model summary). Layers 1-4 were convolutional layers and layers 5-7 were LSTM layers composed respectively of 256, 128, and 64 neurons with dropout rates of 0.5. The final layer was a fully connected Dense layer with one neuron. The activation level of this neuron represented the classification.

The more polarized the activation, the greater confidence that the model had in its classification. Additional details on the choice of this architecture is presented in Appendix C.

Table 1. Model summary for the CNN-LSTM.

Layer	Type	Output Shape	Filters	Kernel Size	Dropout	Recurrent Dropout
0	Input	250 x 12	–	–	–	–
1	1D Conv.	247 x 32	32	4	–	–
2	1D Maxpool	24 x 32	10	1	–	–
3	1D Conv	2 x 32	32	4	–	–
4	1D Maxpool	2 x 128	10	1	–	–
5	LSTM	2 x 256	–	–	0.5	0.35
6	LSTM	2 x 128	–	–	0.5	0.35
7	LSTM	64	–	–	0.5	0.35
8	Dense	1	–	–	–	–
Total parameters		548097				
Trainable parameters		548097				

CNN-LSTM models require data to take the form of a 3D array, wherein the first dimension represents the number of sequences in the dataset, the second dimension represents the number of time steps within each sequence, and the final dimension represents the number of features. Our data took the shape (971, 250, 12). The classifier was trained via 10-Fold grid search cross validation (GridSearchCV; Pedregosa et al., 2011) using Keras with a TensorFlow backend. Each time window consisted of 250 timesteps spaced 75 milliseconds apart thereby covering 18.75 seconds.

3.7.3. Data Preprocessing

The active area in the Tunnel condition was located in a random position on the canvas and all targets appeared in random locations within this region. In the Non-Tunnel condition, targets appeared randomly anywhere on the canvas. Therefore, from trial to trial, even within conditions, behaviour was expected to differ significantly. Furthermore, the direction of successive targets within conditions changed at random. This meant that we were unable to simply pass sequence data (i.e., $t_i - t_{i-1}$) into the classifier. For example, in classifying a data stream from an accelerometer of a user walking up a staircase (e.g., Human Activity Recognition dataset; Anguita, Ghio, Oneto, Parra, & Reyes-Ortiz, 2013), researchers could expect that person, regardless of who the person is or the specifics of that staircase, to be travelling in an

upwards direction (Arif & Kattan, 2015; Vrigkas, Nikou, & Kakadiaris, 2015). Our case is more similar to classifying the activity of playing soccer: The actions should share similar characteristics, but the trajectories of the sequences will differ depending on where the ball is. Figure 11 illustrates how participant behaviour can share general properties but differ depending on the specific locations of the targets.

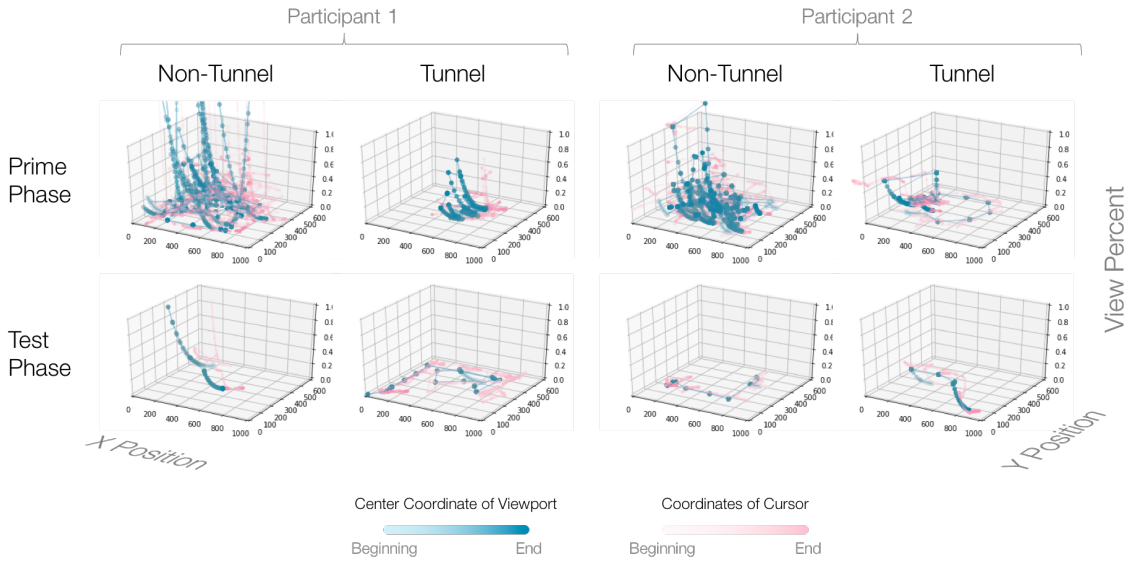


Figure 11. Viewport location and cursor position data from two participants. The X- & Y-axes represent the vertical and horizontal axes from Figure 9. The Z-axis represents the zoom level, thereby illustrating where on the canvas the users were interacting and at what depth those interactions were occurring. As the plots show, in the Tunnel condition the participants exhibited similar patterns of behaviour, but that behaviour was concentrated around different points on the canvas.

To account for this, we calculated how far participants moved from the first recorded point within each sequence passed to the network, irrespective of direction or location on the canvas. This was done independently for each of the three original features described above. This type of preprocessing makes the classifier resilient to the specific location of attentional tunnels.

Because users completed the Tunnel condition significantly faster than the Non-Tunnel condition, we had a roughly 2:1 imbalance in data. We therefore down-sampled the Non-Tunneling data, removing about half of the sequences. We also normalized all data between -1 and 1. Finally, we removed the 13 trials that failed to induce attentional tunnels according to our degree of interaction and performance criteria (see Data Cleaning section).

3.8. Classifier Results

The principal metric for our classifier was mean area under curve (AUC) across the 10-Folds. This is widely recognized as being the best estimator for the performance of a model as it represents an aggregate classification across all classification thresholds (Mason & Graham, 2002). The model achieved a mean AUC of 0.74 across the 10-Folds (see Figure 12). The test dataset was composed of 53% Tunneling observations, which represents chance performance.

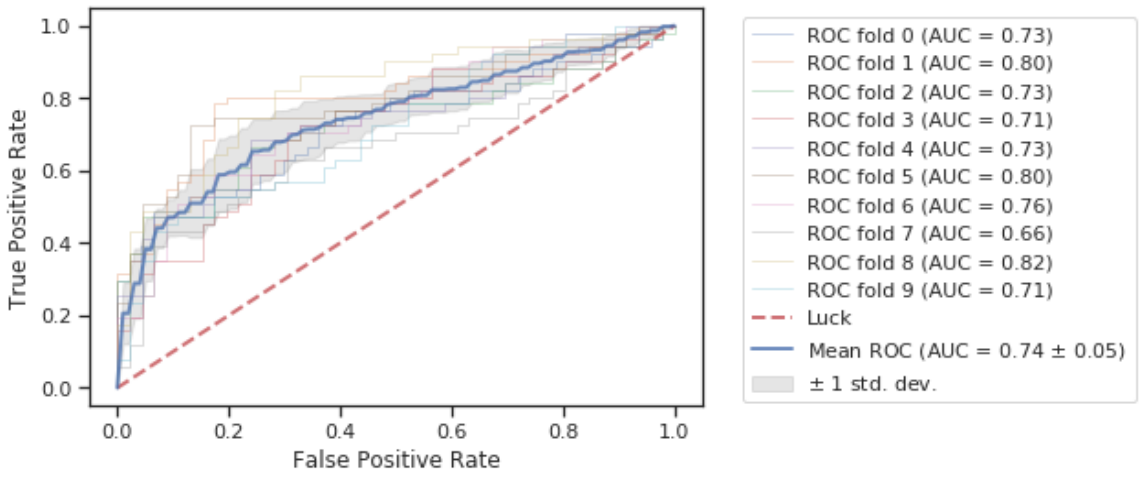


Figure 12. ROC curves for 10-Fold cross validation of the CNN-LSTM model.

3.9. Classifier Discussion

We demonstrated that classification of an attentional state is possible through behavioural metrics, even with a relatively small, noisy, and feature-sparse dataset. This shows promise for adaptive systems based on PDM.

Direct comparison of classifier performance is difficult as there are relatively few benchmarks and the degree to which an attentional state manifests behaviourally is likely contingent on the characteristics of the attentional state itself as well as the task that users are performing while in that state. Our main point of comparison was the Régis et al. (2014) study, as they successfully demonstrated classification of attentional tunnels through largely physiological measures. They achieved higher classification accuracy than we did (91%). However, the differences between experimental conditions were much more overt in their study, so direct comparison is difficult. Mannaru, Balasingam, Pattipati, Sibley, & Coyne (2016) used physiological measures (eye

tracking and pupillary dilation) to measure workload in operators using a desktop task. Again, they achieved higher classification accuracy (88%) but with overt differences between conditions. McDonald et al. (2014) represents an interesting comparison as they also used behavioural metrics, but to infer states of drowsiness in drivers. We achieved a higher classification accuracy than they did as measured by AUC (AUC = 0.70), but because their attentional state and task were very different, it is difficult to compare. These comparisons demonstrate that PDM can yield results that are significantly above chance and comparable to physiological measurement and should therefore be considered when implementing adaptive user interfaces triggered by operator state measurement.

Although the dataset that we trained our classifier on was sufficient to achieve comparable performance, it still had a high degree of variability from participant-to-participant, was relatively small, and included few features. Future studies should examine the performance of similar classifiers trained over longer timespans on a more homogenous participant pool. Such conditions might better approximate those of actual work settings and operator characteristics. Furthermore, as we wanted to evaluate the efficacy of PDM, we only passed interaction data to the classifier. If we were to include system state data or further contextual cues, the accuracy of the classifier would presumably improve. Including these additional features is feasible in many information dense spaces as system state information is often readily available and time-stamped (e.g., cybersecurity; Corchado & Herrero, 2011). Finally, we used relatively short time windows (18.75s). With a sufficiently large dataset, larger time windows could be used, which would also likely increase the accuracy of the classifier.

3.10. General Discussion

This study supports the efficacy of using PDM for inferring attentional states. We were able to validate that the experimental paradigm successfully elicited behaviours that aligned with the Wickens and Alexander (2009) definition of an attentional tunnel while also demonstrating the value of refocusing that definition on the *potential for* rather than the *expectation of* negative outcomes. We then developed a classifier capable of inferring attentional tunnels from data remotely recorded through Amazon's Mechanical Turk. Given the unknown locations of the users, the computers they used to complete the study, and the demographics of the users

themselves, this study demonstrates the potential and practicality of PDM in adaptive automation.

The performance trade-off that we found shows that fixating on a particular area of space should not necessarily be avoided, which indicates that adaptive DSSs shouldn't be designed to "break" attentional tunnels. Rather, depending on the domain, they should be designed to foster an understanding of the stakeholders' attentional processes. For example, a shift supervisor may want to know when their operators are in attentional tunnels so that they can determine the appropriateness of that state. Future studies can also seek to develop automated methods of distinguishing between attentional tunnels and directed attention. Alternatively, the methods described in this paper can be used to determine how susceptible an interface is to inducing attentional tunnels.

To further improve the utility of this classifier in practice, system-state information can be included as either additional features or classifier modulators. For example, Rantanen and Goldberg (1999) demonstrated that increased mental workload narrows an operator's visual field size. If workload was coupled with this classifier, the classification threshold could be modified to facilitate more accurate classifications. Future studies can also examine this research through the lens of the Strategic Task Overload Model (STOM; Wickens, Gutzwiller, & Santamaria, 2015), which may be able to formalize how and when people switch between areas of interest during visual search. This is particularly relevant as attentional tunneling has been cited as a critical aspect of STOM (Wickens & Gutzwiller, 2017).

3.11. Conclusion

Practical applications of adaptive automation have remained sparse due to an inability to accurately capture the context of the adaptations (Feigh et al., 2012). However, recent developments in machine learning may offer a solution to this problem by enabling a richer contextual understanding (Mangos & Hulse, 2017). We show here that combining some of these new methods with PDM can allow for one dimension of attentional context to be understood. Future work should examine the suitability of this approach in combination with different aspects of operational context and for a wide array of attentional states.

Chapter 4 - Information Overload

Phase 2 successfully demonstrated that an enduring attentional state could be detected through machine learning. These techniques could have been used in real-time to introduce interventions aimed at either breaking attentional tunnels or alerting operators or supervisors about when attentional tunnels are occurring. However, as the goal of the project was to develop ALOD displays, we did not incorporate real-time classification in Phase 2, instead using Phase 2 as a proof of concept for PDM.

Phase 3 sought to build off of Phases 1 and 2 by incorporating real-time classification of information overload into an AUI. It comprised Experiments 3.1 and 3.2 (see Figure 1), which respectively identified the behavioural correlates of information overload and implemented a PDM inference engine to drive an AUI that could autonomously calibrate the amount of information that was presented to users. This was the final phase of the dissertation and sought to bring together the methodological contributions of the first two phases. If successful, the results of Phase 3 would demonstrate that a wide array of attentional states are likely to manifest behaviourally and would therefore demonstrate the potential for PDM in practice. Furthermore, it would fulfill the goal of developing an ALOD display for our industry partner.

4.1. Statement of Authorship

This chapter is in review at Human Factors (Kortschot, Jamieson, & Prasad, In Review). It was authored by Sean W. Kortschot, Greg A. Jamieson, and Amrit Prasad. The research questions were conceptualized by Sean W. Kortschot under the supervision of Greg A. Jamieson. Sean W. Kortschot conducted the literature search, created the research design, built both testbeds used for experimentation, and conducted all experimentation through MTurk. Amrit Prasad assisted in the development of the adaptive version of CogLog used in the second experiment of this chapter. Sean W. Kortschot selected, executed, and interpreted all statistical analyses for both experiments and developed the machine learning models used to drive the adaptive interfaces used in the second experiment. This paper was written by Sean W. Kortschot with Greg A. Jamieson providing editorial oversight. Amrit Prasad reviewed and commented on the paper prior to submission.

Detecting and Treating Information Overload with an Adaptive User Interface

4.2. Abstract

Objective: Develop and evaluate an adaptive user interface that can detect states of information overload in an operator and adaptively respond by calibrating the amount of information on the screen.

Background: Machine learning can be used to detect changes in operating context and to trigger adaptive user interfaces (AUIs) to accommodate those changes. Operator attentional state represents a promising aspect of operating context for triggering AUIs. Behavioural rather than physiological indices can be used to infer operator attentional state.

Method: In Experiment 3.1, a network analysis task sought to induce states of information overload relative to a baseline. Streams of interaction data were taken from these two states and used to train machine learning classifiers. We implemented these classifiers in Experiment 3.2 to drive an AUI that automatically calibrated the amount of information displayed to operators.

Results: Experiment 3.1 successfully induced information overload in participants, resulting in lower accuracy, slower completion time, and higher workload. A series of machine learning classifiers detected states of information overload significantly above chance level. Experiment 3.2 identified four clusters of users who responded significantly differently to the AUIs. The AUIs driven by our classifiers improved accuracy, completion time, and workload for three of the clusters.

Conclusion: Behavioural indices can be used to effectively drive AUIs that respond to changes in operator attentional state in some user groups. The success of AUIs may be contingent on characteristics of the user group.

Application: This research applies to adaptive automation aimed at managing operator attentional demands in information-dense domains.

4.3. Introduction

Information overload has been linked to past disasters such as the NASA Challenger shuttle explosion and Iran Air Flight 655 (Fisher & Kingma, 2001). As our ability to gather data increases, the problem of information overload will as well. Eppler and Mengis (2004) define information overload simply as a state where an operator is "receiving too much information" (p. 326). We operationally define it as an attentional load surplus precipitated by the excessive presentation of information.

Concerted design efforts have been made in various information-dense spaces to address information overload. For example, cybersecurity operators are typically tasked with overseeing massive, highly distributed, and dynamic networks, which imposes a state of near-constant information overload (Bennett, Bryant, & Sushereba, 2018; Mitropoulos et al., 2006). Recent efforts to address this problem have included developing ecological interfaces (Bennett et al., 2018), employing virtual worlds (Michel, Helmick, & Mayron, 2011), and focusing on levels of understanding needed to effectively monitor a network (Angelini & Santucci, 2017). Interestingly, Gutzwiller, Ferguson-Walter, Fugate, and Rogers (2018) proposed a method of inverting the multiple resource theory model of attention (Wickens, 2008) to overload attackers, leveraging the information demand on both sides of the battle. Information overload has also been considered in manufacturing (Wu, Zhu, Cao, & Li, 2016), transportation (Dadashi, Golightly, & Sharples, 2017), and healthcare (Carroll et al., 2014; Hall & Walton, 2004). While these efforts have yielded positive results, a parallel effort to increase the amount of available data (Marksteiner, 2009) has caused the problem of information overload to persist (Bennett et al., 2018; Hodgetts, Vachon, Chamberland, & Tremblay, 2017).

The threshold at which the additional presentation of information becomes overload is influenced by a variety of individual and situational factors (Benselin & Ragsdell, 2016; Haase et al., 2014; Misra & Stokols, 2012). Given a heterogenous user population and operating context, any fixed information quantity will therefore be suboptimal for some users in certain contexts. This limitation presents an opportunity for adaptive user interfaces (AUIs), which modify their information content, configuration, or interactions in response to changes in operating context (Feigh et al., 2012). Feigh, Dorneich, and Hayes (2012) list five classes of AUI *triggers*: Environmental, spatio/temporal, task/mission, system, or operator state. Since information

overload is a state of the operator, it can serve not only as the subject for treatment through adaptive design, but also as the trigger for that treatment.

AUIs that respond to individual differences have become a central component of the modern internet. For example, in 2015 Netflix reported that approximately 80% of its total streams were influenced by its personalized recommendation system (Gomez-Uribe & Hunt, 2015). Contrarily, real-time AUIs have not gained the same traction (Feigh et al., 2012). This disparity can be attributed to several factors. First, personalized recommendation systems typically operate over longer timespans, modifying the user interface when it loads thereby masking the adaptations (Gorgoglione, Panniello, & Tuzhilin, 2019). One of the central concerns with real-time AUIs is their potential for obtrusiveness (Feigh et al., 2012; Höök, 2000). A now infamous example of this is Clippy, the Microsoft Office assistant who offered assistance based on inferences of user intent (e.g., writing a letter; Dale, 2016). A second factor is that AUIs have been limited in their capacity to capture the full context in which their adaptations occur (Feigh et al., 2012), which has enabled confounding factors to limit the effectiveness of those adaptations. Finally, different user groups (e.g., age brackets) have been found to respond differently to AUIs (Lavie & Meyer, 2010). This shows that even a well-designed AUI may be detrimental to some users thereby limiting their adoption in the civilian sphere. Although these challenges have limited the effective implementation of AUIs in the past, recent advancements to machine learning (e.g., Mangos & Hulse, 2017) and operator measurement techniques (Kortschot & Jamieson, in press) may offer new avenues for AUI development.

This paper describes two studies focused on detecting and then treating information overload in a simulated network analysis task via an AUI. We begin by discussing our general approach to online measurement of information overload and how these measurements can be used to drive an AUI. We then present two experimental studies whose respective aims were to i) identify the behavioural correlates of information overload for the purpose of developing a machine learning classifier, and ii) use this classifier to drive an AUI capable of online calibration of information quantity.

4.4. Overall Method

4.4.1. Passive Data Monitoring

Using an operator's state to trigger an AUI begins with detecting that state. Feigh et al. (2012) list two methods for measuring operator state: Performance and physiology. While both of these methods have proven to be useful in experimental settings, they face limitations in practice. Performance-based state inference relies on a real time, continuous performance metric, which most real-world domains lack. Physiological state inference requires an initial investment in biometric recording devices as well as users who are willing to both wear those devices and share their biometric data. A third approach is Passive Data Monitoring (PDM), which exploits user interaction data streams that exist irrespective of performance or physiology (e.g., cursor position), and seeks to find patterns in that data that align with cognitive events or states (Kortschot et al., 2018; Palmius et al., 2016). The principal limitation of PDM is that it is limited to domains that are highly interactive and whose data is readily available. Fortunately, many information-dense domains possess these traits (e.g., cybersecurity; Goodall, Lutters, & Komlodi, 2009).

The principal method by which structural properties of PDM data are identified is through machine learning, although some researchers have demonstrated good results using heuristic approaches (see Mac Aoidh, Bertolotto, & Wilson, 2012). The actual classification of patterns in user interactions is relatively straightforward, with some researchers finding good results through random forest approaches (McDonald et al., 2014), while others have demonstrated the viability of deep learning approaches (Kortschot & Jamieson, in press). The more challenging aspect of PDM is in the labelling of the data. Other classification tasks such as image classification have the advantage of near-perfect ground truth in their labelling as well as data that is readily available that belongs to those labels (e.g., Nowak & Rüger, 2010). For example, one can be relatively certain that a picture of an apple is an apple. Furthermore, it is straightforward to generate and label a large corpus of apple images. In contrast, to label streams of user interactions as belonging to a particular attentional state, that state must be first induced and verified within the AUI design context. This process is essentially identical to what is done with physiological state inference. For example, to determine how someone's heart rate responds to stress, you need to observe their heart rate after inducing a stressful state and compare that to

their baseline rate. With PDM, rather than examining a physiological signal's response to an induced state, we examine the response of an interaction signal (e.g., cursor position, scroll behaviour, etc.).

4.4.2. Experimental Platform

The experiments described in this article were conducted on the Cognitive Logger for Information Overload (CogLog_IO), a browser-based platform built with Cytoscape Web (Lopes et al., 2010) that can be deployed through Amazon's Mechanical Turk (MTurk). Both experiments employed the same core experimental task: Users were presented with an artificial subway network and in each trial had to find the fastest trip between an Origin and Destination station. The subway network was created for these studies and consisted of 287 *stations* and 18 *lines* (see Figure 13). A line was composed of a series of *segments* that were connected by stations. Each segment had a speed associated with it, measured in kilometers per hour. These were displayed at the midway point of a segment in text that ran parallel to that segment. The speed of segments along the same line tended to be fairly consistent, but there were sections along lines where the speed changed significantly. Participants were able to switch between lines only at *transfer stations.* Each transfer station had a delay, which was the time it took to transfer from one line to the next. The network was drawn to scale, meaning that travelling along a segment that was twice the length of another would take twice as long, assuming the speeds of the two segments were equal. Finally, each Origin-Destination set had a unique collection of lines and typically one segment that was taken offline that users could not select. These were marked by striped segments, which were difficult to distinguish from non-striped, active segments when fully zoomed out. This demanded additional investigation of the network prior to selecting a path and therefore added further complexity. In summary, participants had to consider the following factors along a candidate trip:

- The speed of line segments,
- the total distance,
- the number of transfers required,
- the delays associated with those transfers, and
- whether any lines or segments were offline.

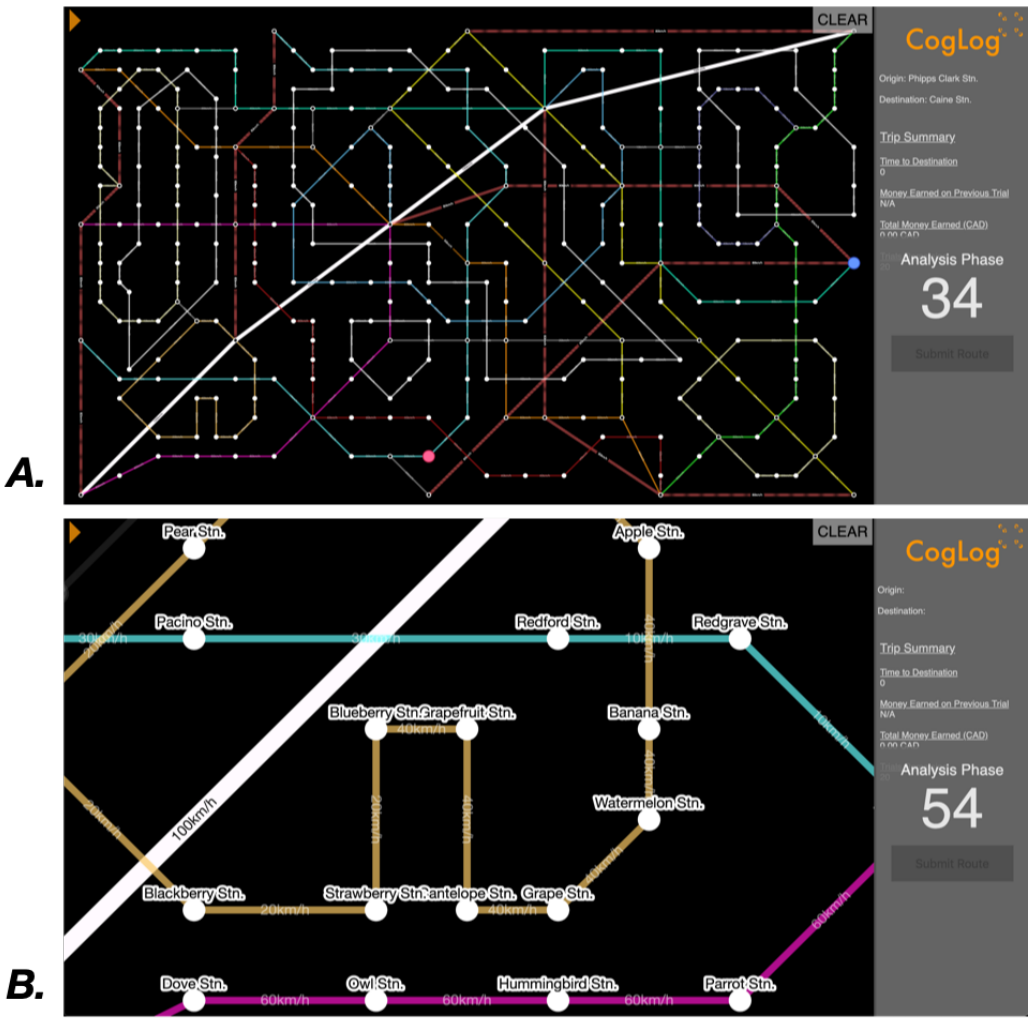


Figure 13. A. Overview of the network used in the experiments. The solid white nodes represent subway stations and the white nodes with a black center represent transfer stations. The Blue node is the Origin station and the red node is the Destination station. These changed each trial but the structure of the network remained the same. B. Sample of the network when fully zoomed-in. Note the visibility of the speeds of the segments. Also note that these figures were originally in colour.

The use of MTurk necessitated additional control measures to ensure active engagement with the task. The first measure was an incentivization scheme whereby participants could earn a CAD 0.25 bonus on each trial if the trip that they submitted was the optimal trip. The bonus was reduced by CAD 0.01 for every four minutes that their submitted trip exceeded the optimal trip until they had lost the full bonus. Participants could earn a maximum of CAD 5.00 over the experiment in bonuses (a 50% increase from the minimum payment in the first experiment and a 33% increase in the second). Second, we included an *analysis phase* at the beginning of each trial, during which segments could not be selected but the network could be investigated. This

phase sought to ensure that participants actively inspected the network rather than either selecting the most obvious trip regardless of the bonus implications or exhaustively attempting different trips to see which resulted in the shortest time. The analysis phase timer would only run when participants were engaged with the platform, thereby discouraging participants from performing secondary tasks during the analysis phase. The duration of the analysis phase timer differed between the two experiments (see relevant Methods sections for justification).

Once a trial began participants were able to see the bonus earned on the previous trial in the sidebar, which allowed them to keep track of their performance during the experiment. The sidebar also displayed their total earnings up to that point in the experiment as well as the number of trials remaining. In Experiment 3.1 the sidebar contained an interactive *minimap*, which was a 200 X 120 pixel scaled representation of the network. It showed where a participant's current viewport fell within the network and allowed participants to jump to new points by clicking on their corresponding minimap location. The minimap was included to allow for additional design opportunities for the AUI in Experiment 3.2. After completing the design process for the AUI in Experiment 3.2 it was determined that the minimap was not necessary and it was therefore removed from the CogLog_IO to reduce the memory demand imposed on participants' browsers.

4.4.3. Platform Interactions

To investigate the network, participants could zoom in by scrolling with their cursor and pan around by clicking and dragging on the network map. Participants were able to select segments by clicking on them, at which point the segment would become bright blue (RGB: 0, 100, 255). CogLog_IO had integrated logic that ensured users began their trip at the Origin station and could only select adjoining segments. The trip could only be submitted when participants had selected a series of segments that linearly connected the Origin to the Destination station. Upon selecting each segment, the time that it took to travel that segment was added to the "Time to Destination" number in the sidebar. If the participant's selection incurred a transfer delay, that delay was also added. This allowed participants to track how long their selected trip would take. Participants could undo their last selection by clicking again on that segment. They could erase all of their selections by clicking the CLEAR button located in the top right corner of the screen.

During experimentation we recorded the user's cursor position relative to both their screen and the network map as well as the size and position of the network map. From this data we derived all panning and zooming behaviour.

4.4.4. Measures

Participant performance was evaluated via three principal metrics: Accuracy, completion time, and workload. Accuracy was measured as *Minutes Over Optimal* (MOO), which was the difference in trip duration between their submitted trip and the optimal trip. Completion time was measured per trial and was calculated as the time between the point that the new Origin-Destination set loaded and the point that participants submitted their trip. Workload was assessed using a NASA-Task Load Index (NASA-TLX; Hart & Staveland, 1988) survey that followed each trial. We combined five dimensions (Mental, temporal, effort, frustration, and performance) to form one overall workload measure for each trial. This excluded the Physical dimension as the experiment did not involve physical exertion. We also performed additional analyses on more nuanced aspects of participant behaviour (e.g., mean zoom level per trial, etc.). These will be defined when they are described in the Results sections.

All research herein complied with the American Psychological Association Code of Ethics and was approved by the Institutional Review Board at the University of Toronto. Informed consent was obtained from each participant.

4.5. Experiment 3.1: Discovery

4.5.1. Motivation

Experiment 3.1 sought to develop a machine learning classifier based on PDM that was capable of distinguishing states of information overload from baseline within the CogLog_IO environment. This required information overload to first be induced in participants relative to a baseline. We then needed to train a machine learning classifier on interactions that were recorded during states of information overload and baseline to teach it patterns that distinguished interactions from the two states.

4.5.2. Methods

4.5.2.1. Participants & Payment

Forty participants were recruited through MTurk. We restricted our recruitment to master workers, who are MTurk workers who have achieved a minimum level of reliability based on previous studies. Participants were paid a baseline rate of CAD 10.00 and could earn up to CAD 5.00 in bonuses. The experiment was expected to take roughly one hour, which included both training and experimentation.

4.5.2.2. Experimental Design

The experiment was a single factor design wherein the experimental manipulation was the method by which participants accessed station information. In the Baseline condition, participants were able to select stations for which they wanted station information by clicking and holding on that station, which revealed the station name, the station type (single or transfer), and if it was a transfer station, the transfer delay. Furthermore, the opacity of down lines was reduced to 0.45. In the Information Overload condition all station details were concurrently displayed and the opacity of all lines remained at 1.0. Figure 14 illustrates an overview of the two conditions.

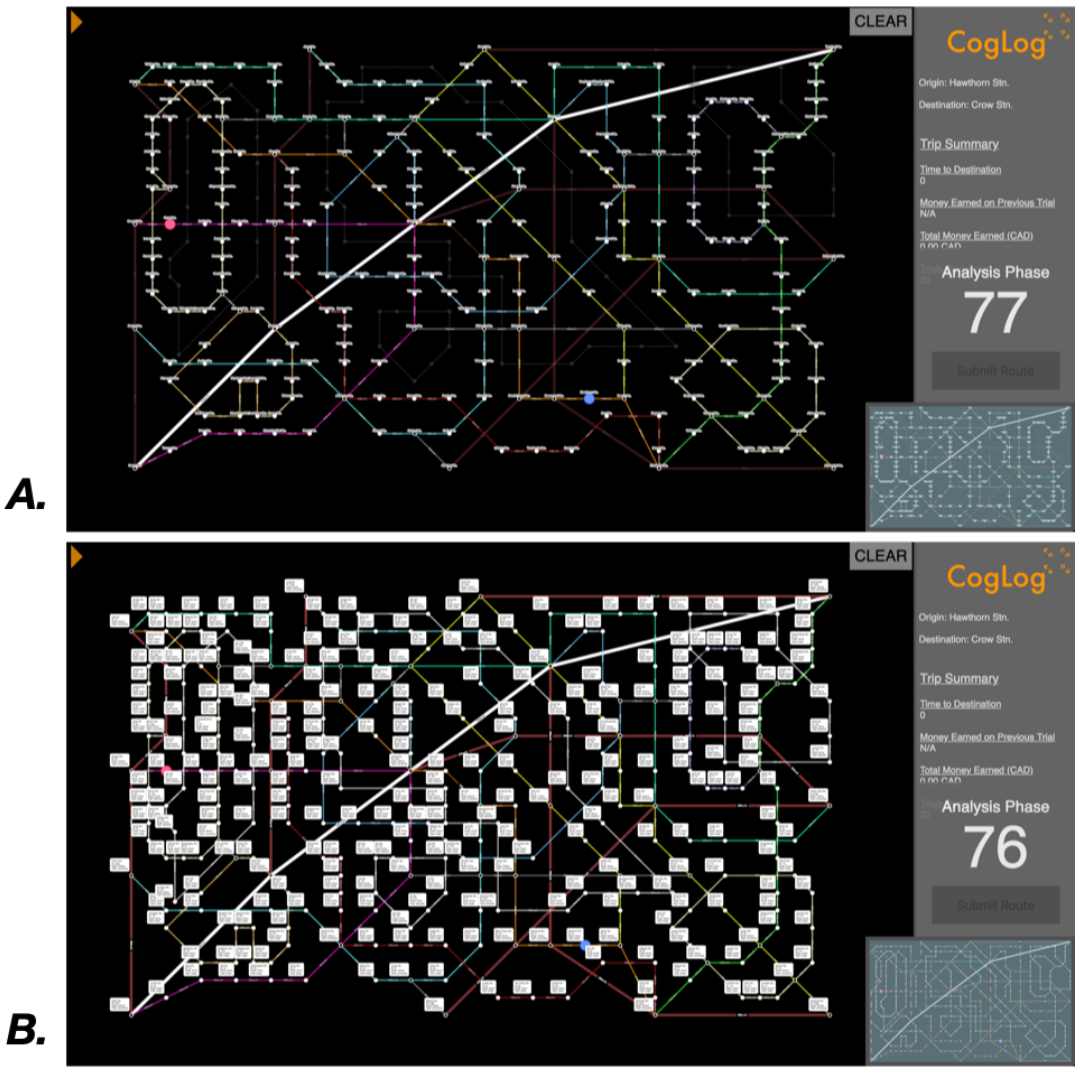


Figure 14. A. Baseline condition wherein users had to click and hold on stations to access the station details. B. Overload condition wherein all station details were concurrently presented. Note that this figure was originally in colour.

We hypothesized that information overload would negatively impact accuracy, but only when completion time was accounted for as a random variable. We also expected completion time and overall workload to increase as a result of information overload.

4.5.2.3. Procedure

Participants were given an overview of the study on the original MTurk request page. They were then directed to a consent form and a screening survey that provided them with a series of nine Ishihara Cards (Ishihara, 1937) that tested for colour blindness. Upon successful completion of the screening survey they advanced to an interactive training program that provided detailed

instructions on the task and the interactions required to complete the task. The program presented a video in one of the two conditions at random of an expert user performing the task. Finally, participants were given two practice trials in each condition. The sequence of these was fully randomized. These trials also included the NASA-TLX questionnaire that participants would receive during experimentation.

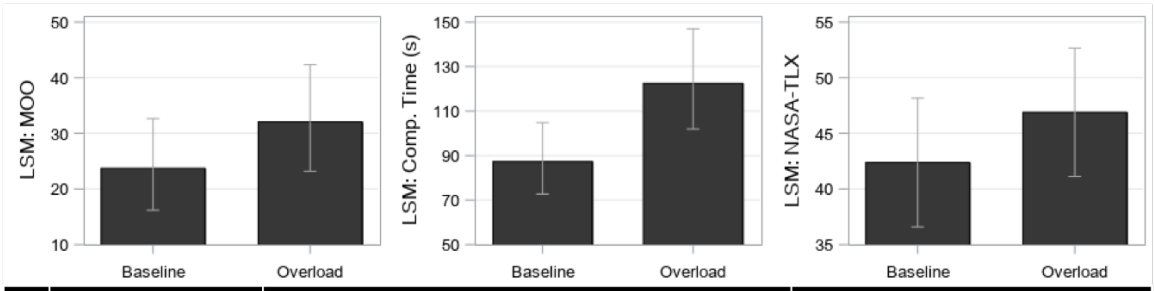
Experimentation involved 10 trials in each condition for a total of 20 trials. Each trial had a unique Origin-Destination set and each set had a designated pair of lines and segments that were taken offline (i.e., not clickable). These were all designed before experimentation to ensure approximate equivalence of difficulty. Both the order of conditions as well as the pairing of Origin-Destination sets with conditions were fully randomized.

The analysis phase at the start of each trial was set to 90 seconds to ensure not only that participants actually searched the network for the best path, but also to ensure that we would capture enough data in each experimental condition to develop a reliable machine learning classifier. Participants were presented with a NASA-TLX following the submission of the trip on each trial.

4.5.3. Results

The remote nature of the experiment exposed our data to the potential for noise caused by participants either taking breaks or engaging in secondary tasks. To control for this, we binned the data into 10 second windows and removed windows wherein the user's cursor position did not change. If a window or series of windows was removed from a trial, that amount of time was deducted from their overall completion time. We identified a total of 106 of these windows amounting to 17.67 minutes of data. We do not claim that this procedure guarantees the remaining data to be entirely representative of participants' active engagement with the task nor do we claim that stagnant 10 second windows were necessarily indicative of task interruptions. These are necessary conceits within our experimental procedure and their validity should be borne out by the success or failure of the classifier. Finally, we subtracted the time spent during the analysis phase (i.e., 90 seconds) from the trial completion time as this was a constant for all trials.

We conducted generalized linear mixed models on accuracy (MOO; see Measures), completion time (seconds), and workload (NASA-TLX). All analyses specified participant, trial number (i.e., learning effects), and Origin-Destination set as random factors. Our model for accuracy also specified completion time as a random factor. The Overload condition yielded significantly lower accuracy, slower completion time, and higher workload compared to the Baseline condition (see Figure 15). Even when completion time was omitted as a random factor, participants achieved significantly better performance in Baseline trials (M = 23.70 MOO, SE = 0.42) compared to Overload trials (M = 32.04 MOO, SE = 0.43; $F(1, 721) = 9.98$, $p = 0.002$).



		Condition LS Means (SE)		Significance
		Baseline	Overload	
Metric	Accuracy *MOO*	23.70 (0.43)	32.04 (0.43)	$F(1, 720) = 9.98, p = .002$
	Completion Time *seconds*	84.12 (0.08)	114.69 (0.09)	$F(1, 720) = 47.59, p < .0001$
	Workload *NASA-TLX*	42.38 (2.95)	46.89 (2.93)	$F(1, 739) = 30.80, p < .0001$

Figure 15. Three principal metrics for Experiment 3.1. LSM refers to Least Squares Means and SE refers to Standard Error. Error bars represent the upper and lower bounds for the LSM predictions.

Finally, we examined mean view percent per trial across experimental conditions. We found that participants in the Overload condition tended to view a significantly smaller portion of the network (i.e., were more zoomed-in; M = 105%, SE = 0.07) compared to those in the Baseline condition (M = 119%, SE = 0.07; $F(1, 721) = 49.50$, $p < .0001$).

4.5.4. Discussion

The results of our first experiment validate that the Overload condition successfully induced information overload in participants relative to the Baseline condition. We showed that increasing the degree of concurrent information presentation, even when that information had the

potential to be useful, resulted in poorer accuracy, longer temporal performance, and increased perceived workload. Since participants had access to the same information in both conditions this result can only be attributed to *how* information was presented rather than *what* information was presented.

The finding that the completion time of trials did not significantly impact participants' accuracy was surprising as we had hypothesized that spending longer on a trial would allow participants to exhaustively search for the optimal trip. A possible explanation for this effect comes from Milgram (1970), who showed that information overload induced states of *attentional tunneling* wherein people focused on information that they perceived to be the most important at the expense of omitting information beyond this scope. Therefore, in states of information overload, participants may have identified an initial candidate trip that they perceived to be optimal and failed to adequately consider alternatives, regardless of how long they spent searching. This explanation is also in line with the findings of Swain & Haka (2000).

A second possible contributing factor is that the additional concurrent presentation of information, although never fully occluding segments, may have obfuscated higher-level structural properties of the network. Essentially, by including a surplus of micro-level details, participants could have missed the macro elements of the network. This is supported by our finding that participants tended to be significantly more zoomed-in during Overload trials.

4.6. Experiment 3.1: Machine Learning Classifier

4.6.1. Motivation

The goal of this stage of our study was to develop a machine learning classifier capable of identifying whether streams of interaction data belonged to a state of information overload or not.

4.6.2. Methods

We restricted our classification approaches to relatively simple random forest and decision tree classifiers so that they could be implemented on the front end of CogLog_IO using basic JavaScript. CogLog_IO sampled interaction features every 200 milliseconds, which resulted in 763350 samples covering approximately 42 hours. We divided the dataset into 10 second

windows and passed each window independently to the classifier. The data consisted of the X and Y coordinates of the user's cursor relative to the screen, their screen's size, the position of the network map, and the size of the canvas. From this data we were able to derive the zoom level, the location of the user's cursor relative to the network, and the portion of the network that was currently in view. We were also able to transform this data to derive all panning, zooming, and cursor movement behaviour.

The subway task from CogLog_IO imposes a significant limitation on classification in that the directions of the input vectors are significantly influenced by the arbitrary nature of the Origin-Destination set locations. For example, moving the cursor from the top-left to the bottom-right of the map is not necessarily indicative of the attentional state from which those interactions were captured. Rather, it is more likely that the Origin was located in the top-left of the map and the Destination was located in the bottom-right. This is not the case for all behavioural classification tasks (e.g., Human Activity Recognition; Vrigkas, Nikou, & Kakadiaris, 2015). To control for this, all inputs were converted to a representation that considered only the magnitude of movement and not the direction. This was done by calculating the absolute distance between the first observation in each 10 second window and each subsequent observation within that window. Therefore, for all transformed features, the first observation in each window was set to 0.00. If that input stream didn't change, the values remained at 0.00. If the input stream moved away from the initial value in any direction, the value increased. If during that 10 second window the variable moved closer to the starting point, the value decreased. Figure 16 illustrates an example transformation of a user's cursor position for two consecutive 10 second windows.

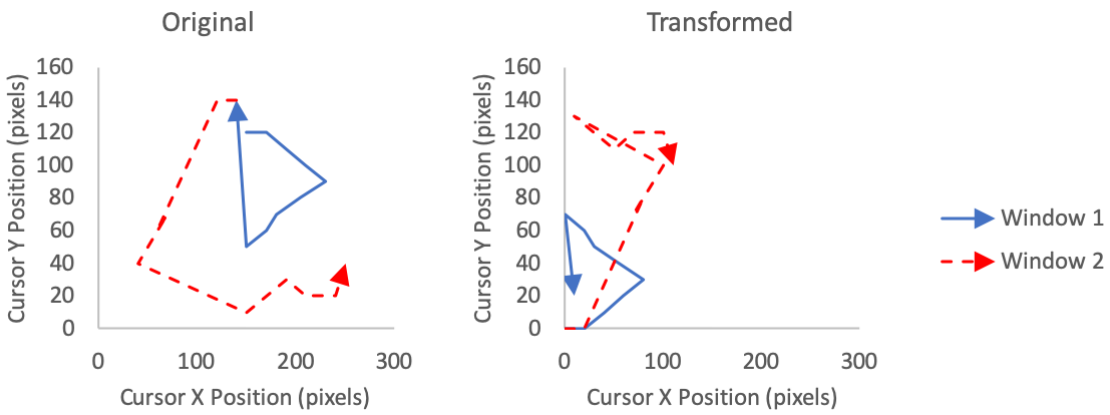


Figure 16. Example of data transformation for cursor position. Note that there was an additional transformation that combined the X and Y magnitudes for cursor position.

Feature extraction was accomplished using tsfresh (Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests; Christ, Braun, Neuffer, & Kempa-liehr, 2018). Again, because we sought to demonstrate the practicality and relative ease of this approach in real-world settings, we restricted our features to simple features, which are those that produce single number outputs (e.g., mean zoom over 10-second window).

4.6.3. Results

We ran tsfresh on two sets of input features. The first included the raw view percent feature as well as the magnitude-transformed features of the cursor position relative to the canvas, cursor position relative to the screen, and the position of the canvas. We observed that the classifier trained on this feature set predominantly focused on view percent. Therefore, our second run replaced the raw view percent feature with a magnitude-transformed view percent, which captured the dynamism characteristic of zooming in and out.

We then ran a random forest classifier with the extracted features from both of the tsfresh runs. The random forest classifiers were conducted using Scikit-Learn's random forest module (Pedregosa et al., 2011) with 300 estimators and no maximum depth. We extracted the six most important features from the random forest models and passed them into simple decision trees whose depths were limited to five layers. Again, we developed the decision trees using Scikit-Learn's machine learning module. Table 2 presents confusion matrices summarizing the results from these runs and shows that all four classifiers performed above chance (0.5).

Table 2. Confusion matrices for the two random forest and decision tree models.

			Raw View Percent		Transformed View Percent	
			Predicted Label			
			IO	Baseline	IO	Baseline
Random Forest	True Label	IO	1675	67	1386	356
		Baseline	122	1190	790	522
		Precision	*93*		*63*	
		Recall	*96*		*79*	
Decision Tree	True Label	IO	1705	37	1210	532
		Baseline	833	479	724	588
		Precision	*67*		*62*	
		Recall	*97*		*69*	

4.6.4. Discussion

These results demonstrate that information overload in CogLog_IO manifests through behavioural patterns that are detectable using simple machine learning techniques with a noisy, heterogeneous participant pool.

The principal feature that characterized states of information overload was view percent, which was a measure of the amount of the network map that was within the participant's viewport (i.e., zoom level). This was demonstrated by the decrease in both precision and recall (see Table 2) when moving to a model that only considered the magnitude-transformed representation of view percent. It is therefore likely that subjects tended to zoom in and remain zoomed-in during states of information overload. This is in line with expectations, as zooming in on a smaller portion of the network essentially reduced the total amount of information within view. Furthermore, this supports past findings that information overload could induce states of attentional tunneling in participants (Milgram, 1970; Swain & Haka, 2000).

These results support the use of PDM for inferring operator's attentional states. Using purely interaction data, a limited feature set, and simple machine learning techniques, we were able to successfully classify the operator's attentional state. Future research should augment these models with additional contextual cues outlined in Feigh, Dorneich, and Hayes (2012) or Eppler and Mengis (2004).

4.7. Experiment 3.2: Intervention

4.7.1. Motivation

Our first experiment successfully induced states of information overload in participants. We were then able to use behavioural streams from these states to develop a machine learning classifier that performed well above chance level when classifying states of information overload versus baseline. Experiment 3.2 sought to evaluate the effectiveness of an AUI driven by the two learned decision trees from the first experiment relative to experimental controls.

4.7.2. Design of the AUI

To successfully perform in CogLog_IO, a user must understand both low-level details of stations and lines (i.e., speeds, transfer times) as well as the higher-level geography of the network. Our AUI aimed to facilitate these different levels of understanding, taking inspiration from Sarkar and Brown's (1992) fisheye design (see also Furnas, 1986), which distorted a portion of an underlying image such that an area of interest was magnified relative to peripheral regions. This design allows localized detail to be displayed without sacrificing the context of the surrounding areas (Cockburn, Karlson, & Bederson, 2009), assuming a trade-off between these two levels of detail.

Our final design was not a traditional fisheye design in that there was no distortion involved. Instead, station information within a radius (the *focal radius*) of the user's cursor position was displayed. This allowed users to view the delays at stations within this radius without losing the higher-level geometry of the surrounding network. This design assumes that in some instances, peripheral information will be useful, but only if the user is able to tolerate it. To determine the amount of peripheral information a user was able to tolerate, we implemented our classifiers. CogLog_IO examined how the user interacted with the network over the previous 10 seconds (with a sliding window) and automatically calculated all of the features that were included in our classifiers. It passed these features to the classifiers, which then reported a classification of either "Information Overload" or "Not Information Overload" as well as the confidence it had in that classification. These confidences were then used to increase or decrease the size of the focal radius proportionally with greater confidence resulting in a greater change to the focal radius.

This design sought to automatically calibrate the amount of information to the particular users' information tolerance. Figure 17 illustrates an example workflow for these adaptations.

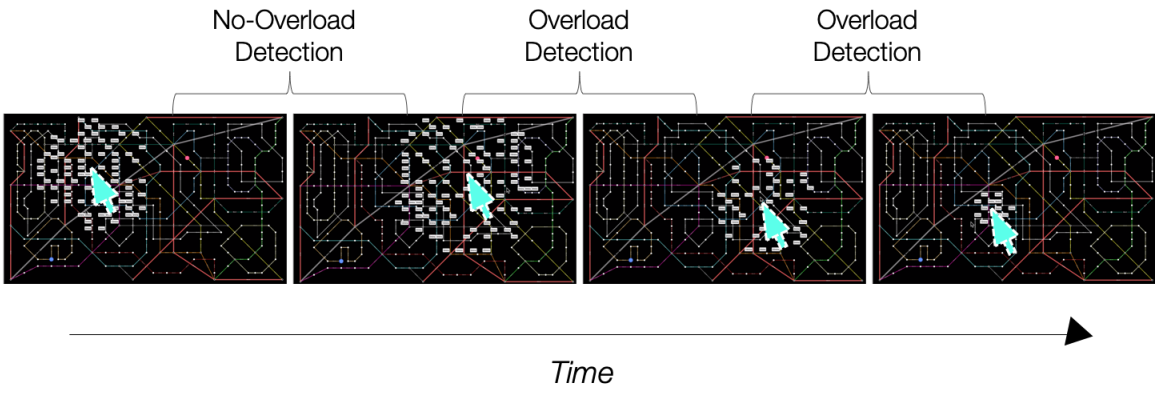


Figure 17. Timeline of how the focal radius changes in response to Overload/Non-Overload detections. Note that the size and colour of the cursor have been changed for illustrative purposes. Also note that this figure was originally in colour.

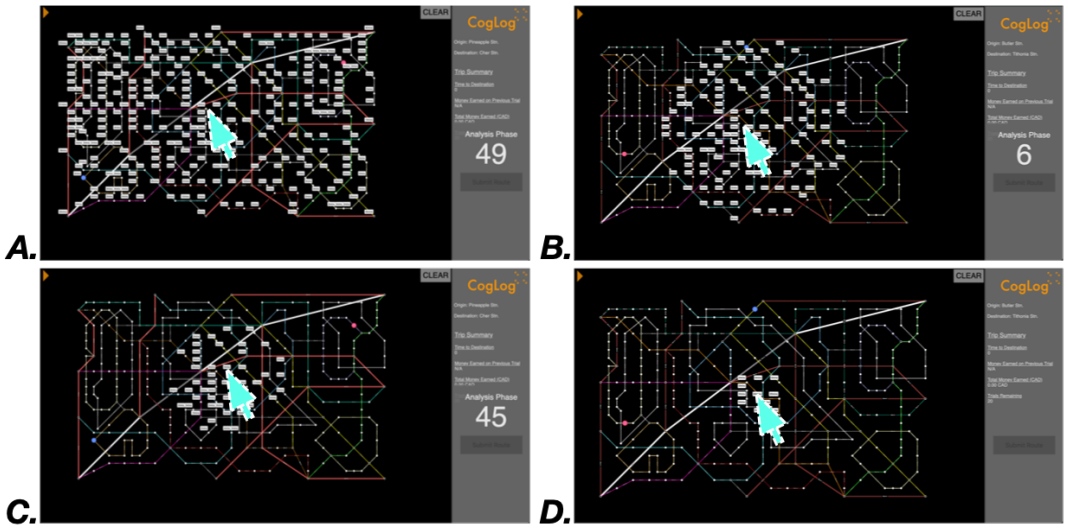
4.7.3. Methods

4.7.3.1. Participants & Payment

Forty participants were recruited through MTurk. We prevented anyone who completed our first experiment from completing the second as we used the same Origin-Destination sets. Since the training was slightly longer in this experiment, we increased the baseline payment to CAD 15.00 while keeping the incentivization scheme the same (i.e., maximum bonus per trial: CAD 0.25).

4.7.3.2. Experimental Design

We employed a single factor design with five experimental conditions. Three of these conditions (*Static*, *Fixed*, & *Random*) were controls and the other two (*DT1* & *DT2*) were variants of our adaptive interface whose difference was the classifier driving the adaptations. Figure 18 presents an overview of the experimental conditions. The opacity of the segments as well as their corresponding speeds was displayed in the same way in all conditions. The only difference between conditions was therefore how the transfer delays were presented.



	Class	Condition	Description
Static	The transfer delay for every station in the network was concurrently displayed	Static	No changes to information bubbles (see A.)
Dynamic	The transfer delays for stations within the focal radius of the user's cursor were concurrently displayed	DT1	Classifier did not include raw View Percent feature. Focal radius changed in size with B. and D. being set to the upper and lower size limits, respectively.
		DT2	Classifier included raw View Percent feature. Focal radius changed in size with B. and D. being set to the upper and lower size limits, respectively.
		Fixed	Focal radius was held constant in size at C. but still followed cursor position
		Random	Focal radius changed in response to a noisy Sine wave with B. and D. being set to the upper and lower size limits, respectively.

Figure 18. Experimental conditions used in Experiment 3.2. Note that this figure was originally in colour.

The Static condition was meant to serve as a baseline against which all other conditions would be evaluated and represented a state likely to induce information overload similar to that from Experiment 3.1. The Fixed condition was designed to evaluate whether any measurable differences in performance or workload could be attributed to the fact that the focal radius followed the user's cursor. The Random condition was designed to control for the possibility that participants were responding to the fact *that* the radius was changing rather than *how* the radius was changing. By including these three controls we sought to isolate the effects of the PDM-driven adaptive designs.

Overall, we hypothesized that DT2 would yield the best accuracy, completion time, and workload, as it employed the most accurate decision tree from Experiment 3.1. We expected this to be followed closely by DT1 and then by the Fixed condition. We hypothesized that the Random condition would be poorer than the Static condition, as we expected participants to find the adaptations obtrusive. At a more granular level, we anticipated that the better overall performers would be more accepting of the Dynamic conditions whereas poorer performers would prefer the Static interface.

4.7.4. Procedure

Participants underwent an identical consent and screening procedure, and a nearly identical training procedure as the one described in Experiment 3.1 (see Experiment 3.1 - Procedure). The central difference was that participants completed only one practice trial in each condition. The ordering of these trials was fully randomized as was their pairing with the Origin-Destination sets used for training.

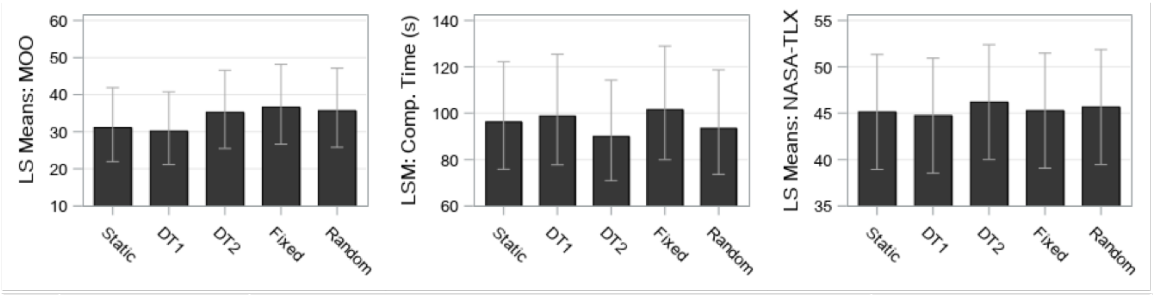
Participants completed four trials in each of the five conditions over the course of the experiment, each with a different Origin-Destination set. These sets as well as their corresponding down lines and segments were identical to those used in Experiment 3.1, as was the process by which they were randomized. Once again, each trial was followed by a six question NASA-TLX.

We reduced the analysis phase timer to 60 seconds as we were no longer concerned with collecting a sufficiently large dataset for developing a machine learning classifier. Through pilot testing we determined that 60 seconds was sufficient for participants to effectively engage in route planning behaviours, thereby giving the adaptive interfaces an opportunity to facilitate the task as intended. There was no upper limit on the time that participants could spend selecting a trip, so if sixty seconds was insufficient, they could simply continue searching for the optimal trip.

4.7.5. Results

We employed the same data cleaning procedures as outlined in Experiment 3.1 (i.e., identify stagnant 10 second windows; see Experiment 3.1 - Results). This removed a total of 95 windows amounting to 15.83 minutes of data.

We conducted generalized linear mixed models on Accuracy (MOO), completion time (seconds), and workload (NASA-TLX), specifying condition as the fixed factor and participant, trial, and route as random factors. In the accuracy model, we also included completion time as a random factor. We did not find any significant differences between the five experimental conditions for any measure. Figure 19 presents a summary of these models and statistics.



		Condition LS Means (SE)					Significance
		Baseline	DT1	DT2	Fixed	Random	
Metric	Accuracy *MOO*	31.11 (0.21)	30.17 (0.21)	35.26 (0.20)	36.61 (0.21)	35.68 (0.21)	$F(4, 717) = 0.71, p = 0.58$
	Completion Time *seconds*	96.27 (1.13)	98.78 (1.13)	90.02 (1.13)	101.55 (1.13)	93.47 (1.13)	$F(4, 718) = 0.71, p = 0.58$
	Workload *NASA-TLX*	45.14 (3.16)	44.74 (3.16)	46.20 (3.15)	45.28 (3.16)	45.66 (3.16)	$F(4, 717) = 0.47, p = 0.76$

Figure 19. Summary of principal statistical findings from Experiment 3.2.

We found a significant effect of condition on mean view percent ($F(4, 718) = 6.43$, $p < .0001$), showing that the static condition ($M = 135\%$, $SE = 0.12$) caused participants to zoom in more than all other conditions (DT1: $M = 155\%$, $SE = 0.12$; DT2: $M = 156\%$, $SE = 0.12$; Fixed: $M = 160\%$, $SE = 0.12$; Random: $M = 160\%$, $SE = 0.12$).

In an effort to uncover whether specific user groups responded differently to the experimental conditions, we performed cluster analysis on the combined results of accuracy, completion time, and workload. Cluster analysis is a common method for identifying user groups within a broader population (e.g., Chan, Cho, & Novati, 2012). We conducted generalized linear mixed models that were able to account for factors like experience and Origin-Destination set. We took the predictions of the individual models for accuracy, completion time, and workload, and standardized them within each participant to have a mean of 0 and a standard deviation of 1. We added the three measures together to form an Overall Decrement Scale. Decrement was used

because higher values for all three of the metrics indicated worse performance or workload. This created a representation of each participant's relationship to the experimental conditions in terms of the relative values to one another. Figure 20 shows a simplified workflow of our clustering methodology.

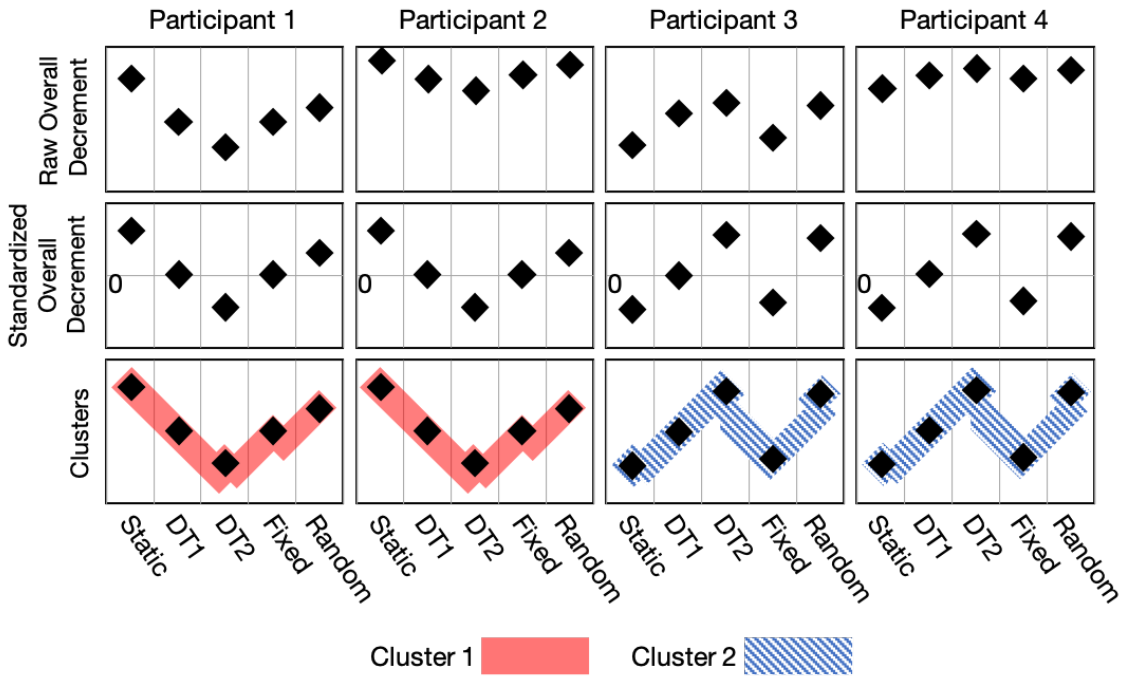


Figure 20. Example of the process by which we clustered participants. Without individual standardization Participants 2 and 4 may have been clustered together based on Overall Decrement rather than their relationship to the experimental conditions.

We used the Lance-Williams flexible Beta method of clustering (Lance & Williams, 1966) with a Beta value of -0.80 to form four clusters, which are depicted in Figure 21. We labelled the clusters according to the condition or class of conditions (i.e., static or dynamic) that yielded the greatest overall benefit: Pro-Decision Trees (Pro-DT; n = 8), Pro-Static/Decision Trees (Pro-Stat./DT; n =6), Pro-Dynamic (Pro-Dyn.; n = 11), and Pro-Static (Pro-Stat.; n = 15). Within each cluster there was a significant effect of condition on Overall Decrement (Pro-DT: $F(4, 28) = 46.67$, $p < .0001$; Pro-Stat./DT: $F(4, 20) = 28.95$, $p < .0001$; Pro-Dyn.: $F(4, 40) = 2.59$, $p = .05$; Pro-Stat.: $F(4, 56) = 10.09$, $p < .0001$).

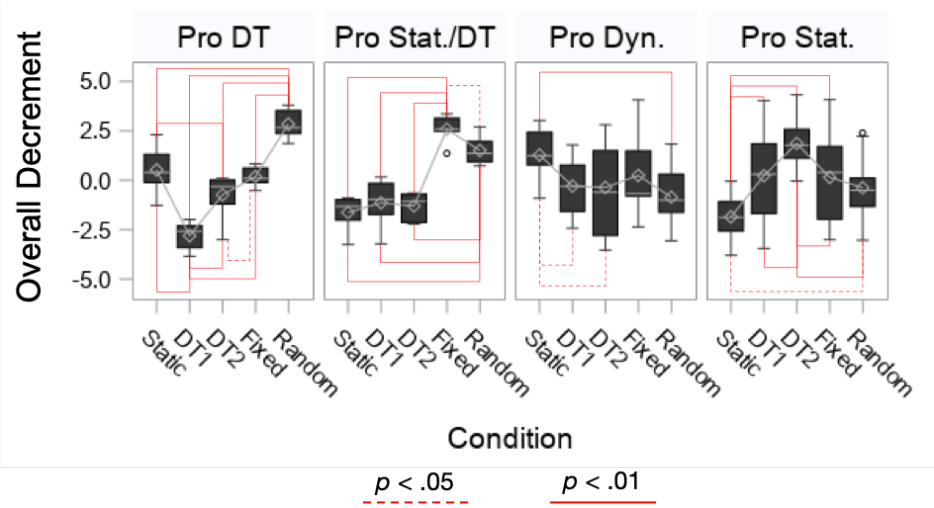
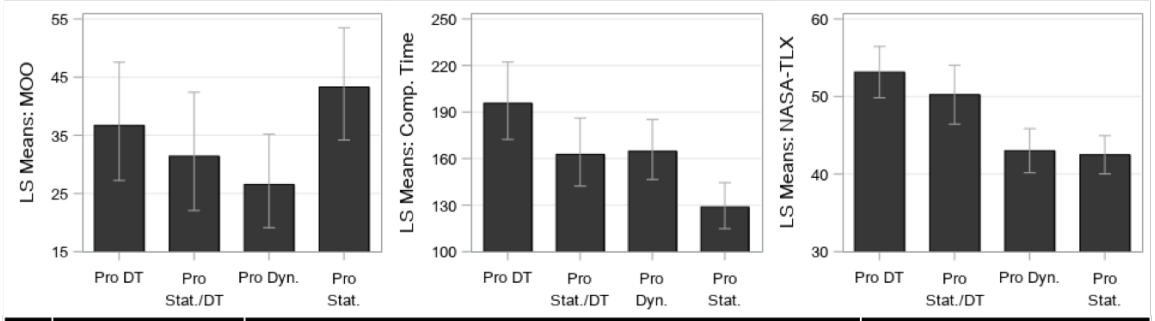


Figure 21. Overall Decrement measures for the four clusters of participants. Lower scores indicate greater benefit derived from that experimental condition. Significant differences are shown via the red brackets and their significance levels are shown via the line pattern.

We evaluated whether there were characteristics beyond participants' relationships to the experimental conditions that distinguished participants between clusters. For example, did participants who tended to perform better in the dynamic conditions also tend to be the best performers overall? We conducted generalized linear mixed models on accuracy (MOO), completion time (seconds), and workload (NASA-TLX), specifying condition, trial number, and Origin-Destination set as random factors. For the accuracy model, we also specified completion time as a random factor. As Figure 22 illustrates, there were significant differences for all three metrics across participants from the four clusters.



		Cluster LS Means (SE)				Significance
		Cluster 1	Cluster 2	Cluster 3	Cluster 4	
Metric	Accuracy *MOO*	36.70 (0.18)	31.41 (0.21)	26.55 (0.16)	43.30 (0.14)	$F(3, 753) = 24.36, p < .0001$
	Completion Time *Seconds*	195.69 (1.07)	162.74 (1.07)	164.75 (1.06)	128.79 (1.06)	$F(3, 753) = 6.22, p < .001$
	Workload *NASA-TLX*	53.14 (1.68)	50.24 (1.93)	43.01 (1.45)	42.49 (1.26)	$F(3, 754) = 12.31, p < .0001$

Figure 22. Accuracy, completion time, and workload metrics for the four clusters.

To determine the overall performance of each cluster we standardized accuracy, completion time, and workload to have a mean of zero and a standard deviation of one. We added a constant of two to each measure such that there were no negative values and performed a cube-root transformation. We then looked at overall decrement by trial across clusters using a generalized linear mixed model with condition, trial, and route as random factors. We found a significant effect ($F(3, 754) = 9.90$, $p < .0001$) overall with the only non-significant pairwise comparison being the difference between the Pro-Dyn. and Pro-Stat. ($t(754) = 0.21$, $p > .05$) clusters. We also repeated the analysis using only the combined results of workload and accuracy since participants were never given explicit instruction to complete the task as fast as possible. This revealed a similarly significant effect ($F(4, 754) = 8.65$, $p < .0001$), but showed that the Pro-Dyn. cluster ($M = 1.52$, $SE = 0.02$) included significantly better performers than the Pro-Stat. cluster ($M = 1.56$, $SE = 0.02$; $t(754) = 2.06$, $p = .04$). Figure 23 presents the overall decrements by cluster for the two tests.

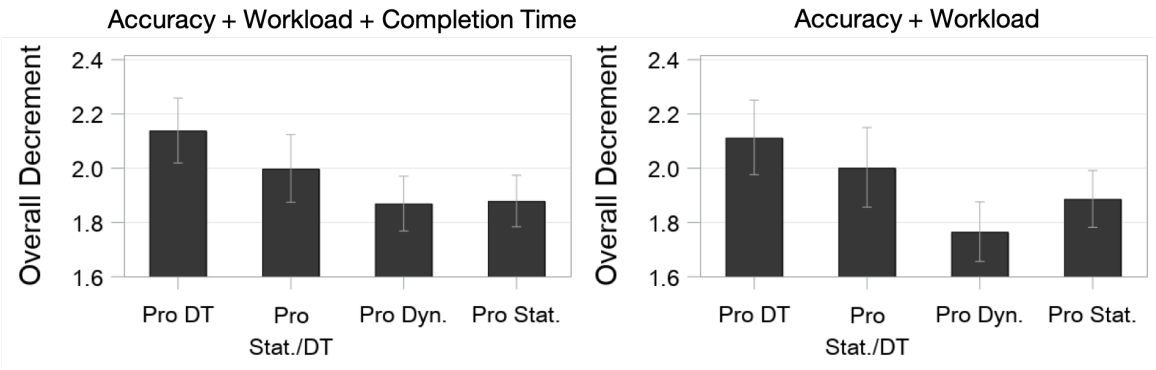


Figure 23. Least Squares Means for Overall Decrement by cluster.

Finally, we examined the economy of participants across the four clusters. We found a significant effect of cluster on economy ($F(3, 754) = 19.55$, $p < .0001$). Pairwise comparisons revealed the Pro-Stat cluster as significantly more economic (M = CAD 24.37/h, SE = 1.16) than the Pro-Dyn. cluster (M = CAD 21.91/h, SE = 1.20; $t(754) = 3.18$, $p < .01$), the Pro-Stat./DT cluster (M = CAD 20.88/h, SE = 1.31; $t(754) = 3.70$, $p < .001$), and the Pro-DT cluster (M = CAD 17.95/h, SE = 1.25; $t(754) = 7.53$, $p < .0001$). We also found that participants in the Pro-Stat./DT and Pro-Dyn. clusters were significantly more economic than those in the Pro-DT cluster ($t(754) = 2.79$, $p < .01$; $t(754) = 4.38$, $p < .0001$).

4.7.6. Discussion

Our initial analyses revealed no significant differences between the experimental conditions in terms of completion time, accuracy, or overall workload. However, we found considerable variance within conditions, indicating that the effect of the experimental manipulation may not have been consistent across participants. Cluster analysis identified four clusters of participants: Pro-Decision Trees (Pro-DT), Pro-Static /Decision Trees (Pro-Stat./DT), Pro-Dynamic (Pro-Dyn.), and Pro-Static (Pro-Stat.). Participants within clusters exhibited a similar relationship to the experimental conditions (see Figure 20). However, we observed considerable differences between clusters not only in terms of how their constituent participants experienced the experimental conditions, but also in terms of overall performance and workload metrics. This clear segmentation partially explains the initially null results and may shed light on the promise and pitfalls of AUIs in practice.

The best performing cluster overall was the Pro-Dyn. cluster, which contained participants who exhibited significant aversion to the Static condition relative to all Dynamic conditions. These participants achieved the highest accuracy, completed the task relatively quickly, reported low workload scores, and achieved the second highest levels of economy. Surprisingly, these participants exhibited no difference in Overall Decrement across the four Dynamic conditions, demonstrating a resilience to the nuances of the adaptive interfaces. This shows that for the highest performing participants, all of the Dynamic conditions helped to manage the information demand imposed by CogLog_IO.

Participants in the Pro-Stat./DT cluster showed preference for the Static condition and the two DT conditions, which shows that these participants only preferred dynamism if it was driven by PDM. These participants achieved the second highest levels of accuracy and relatively fast completion times, but reported significantly higher levels of workload than the Pro-Dyn. participants. They also achieved the same level of economy as those in the Pro-Dyn. cluster, which shows that these participants were likely engaged with the task, but generally found it more challenging than participants in the Pro-Dyn. cluster.

We believe that the low accuracy and workload, fast completion time, and high economy exhibited by participants in the Pro-Stat. cluster demonstrated a prioritization of speed over accuracy. This suggests that these participants may have realized that efficiency over accuracy represented a more economic approach to the experiment and were likely not very engaged with analytic behaviours, instead opting for the most obvious or direct trip from Origin to Destination. Unfortunately, economy is one of the principal motivations for MTurk workers (Hauser & Schwarz, 2016). This cluster may therefore represent an artifact of the experimental sample rather than being representative of an expert user group intent on achieving the highest levels of performance. Future research can employ an incentivization scheme that allocates a greater percentage of the overall earnings to performance incentives.

The Pro-DT cluster also exhibited considerable sensitivity to the dynamics of the interface that they were using, showing a strong overall benefit derived from DT1 relative to all other dynamic conditions as well as the static condition. They also exhibited the second lowest Overall Decrement from DT2. These participants tended to be the worst performers, completing the task the slowest without achieving significantly improved performance, which resulted in the lowest

economy out of any cluster by a significant margin. This result is surprising, as we had hypothesized the more proficient users would show a stronger affinity towards the AUIs that were driven by learned decision trees. However, it is possible that this user group had a lower information overload threshold. This would have resulted in more information overload classifications and thus the tendency for DT1 and DT2 to minimize the amount of information on the screen relative to all other conditions. This lower information overload threshold may be reflective of this user group being less familiar with the type of task that CogLog_IO presents. These findings show that AUIs driven by PDM may be well suited to accommodating a more novice user group. Future studies should explicitly examine how users with different levels of expertise respond to AUIs.

We also found a significant effect of condition on mean view percent with participants in the static condition zooming in more than in all other conditions. This supports the findings from Experiment 3.1 and shows that participants sought to manage information overload by zooming in on a smaller portion of the network. This also demonstrates that the dynamic conditions tended to better manage the information demand imposed by CogLog_IO.

Although these results do not demonstrate a clear improvement derived from AUIs driven by a PDM inference system, they do demonstrate that both PDM and AUIs have the potential to significantly improve performance and decrease workload for some users. This sheds light on not only why these interfaces have had difficulty gaining traction in applied spaces (Feigh et al., 2012), but also their potential in these spaces if implemented in a way that aligns with the needs of that user group. The best performing cluster achieved significantly lower Overall Decrement in all Dynamic conditions. This suggests that expert users are likely to be receptive to AUIs. Conversely, we also found two clusters who tended to be the poorest performers significantly preferred the AUIs driven by PDM inference systems, which suggests that PDM may be also well suited to assisting information demand imposed on a novice participant pool.

4.8. General Discussion

Taken together, the results from the two experiments demonstrate the potential for AUIs driven by PDM. We showed that excessive concurrent presentation of information induced a state of information overload and that this state presented itself in predictable and consistent patterns of behaviour. We then implemented a simplified version of these classifiers as the engine for an

AUI and showed that, for certain users, these interfaces yielded improvements to performance and workload. For users who were the least engaged with the task however, they offered minimal value. These results support the continued development and evaluation of AUIs in applied spaces. They also reveal important shortcomings limiting the widespread adoption of AUIs.

Feigh et al. (2012) state that "adaptive systems have remained limited because of the difficulty in assessing context…" (p. 1009), a point initially highlighted by Bainbridge (1983). Recent advancements to machine learning present the potential to address this deficiency by allowing for systems that are able to form a more robust understanding of their operating context (Mangos & Hulse, 2017). This research used some of these methods and showed that performance and workload can be improved by considering a single aspect (i.e., information overload) of a single contextual element (i.e., operator attentional state). It is likely that a multifaceted context is necessary to realize the full potential of AUIs. Future research should evaluate the efficacy of AUIs that consider a broader context.

There is a growing body of literature suggesting the viability of PDM as a mechanism for operator state inference (e.g., Kortschot et al., 2018; McDonald et al., 2014; Palmius et al., 2016). However, the body of literature wherein AUIs driven by PDM are actually implemented is much more limited. Mac Aoidh et al. (2012) represents the closest point of comparison to the present research as they demonstrated that user interactions can be leveraged to infer states of information overload in a map-based task. However, they did not implement an AUI based on this inference mechanism. Kortschot and Jamieson (in press) showed that attentional tunneling manifests in predictable patterns of behaviour that could be identified via deep learning approaches and describe the framework for PDM, but also did not implement an AUI based on their inference technique. This research therefore represents an important proof of concept for PDM as the driver for an AUI in practice.

There were several limitations that likely influenced the efficacy of our AUIs. The first is that we exploited a relatively sparse feature set in our classification approach, limiting tsfresh to only simple features and then only selecting the top six from that list. More complex features can facilitate the identification of more nuanced structural properties of the data, which may be important in different tasks or for different attentional states. A second key limitation is that we restricted our machine learning inference mechanism to relatively simple decision trees. We

believe that implementing more complex deep learning approaches such as Long Short-Term Memory neural networks (Hochreiter & Schmidhuber, 1997) would likely result in more precise detection of information overload, even with a limited feature set (Lipton et al., 2015; Lipton, Kale, Elkan, & Wetzel, 2016). Third, our use of MTurk also resulted in a participant pool that was presumably more diverse than would be typical in many applied settings. Therefore, a classification rule learned from one participant may not have been well suited to another as the two participants may have approached the task in different ways. Finally, participants in Experiment 3.2 received limited exposure to the different interfaces. This is particularly problematic due to the novelty of the task and interfaces. Furthermore, participants switched between conditions on most trials, thereby requiring constant updating of their mental models of the user interface. Future research should examine longitudinal experience with an AUI.

4.9. Conclusions

As technology permeates more areas of everyday life it will be increasingly important for it to accommodate the changing mental needs of its users. In applied spaces, where the amount of available data vastly exceeds human processing capabilities, this may be especially important. Recent developments in machine learning offer promising solutions to some of the key limitations that have hindered the effective implementation of AUIs in the past. This work demonstrated the viability of using some of these techniques to manage the information demand imposed on users. However, it also showed that users respond differently to AUIs, which suggests that future adaptive systems will need to consider a greater context to realize their full potential.

Chapter 5 - Conclusions

5.1. Summary of Main Findings

The research outline presented in the introduction describes two progressions spanning three phases of experimentation. For the sake of readability, Figure 24 presents the same outline. Both progressions were aimed at developing and evaluating PDM for use in adaptive systems. The first progression sought to move from detecting attentional events to attentional states. The second sought to move from offline, heuristic identification and intervention of cognitive bottlenecks to online, machine learning-based identification and intervention of cognitive bottlenecks.

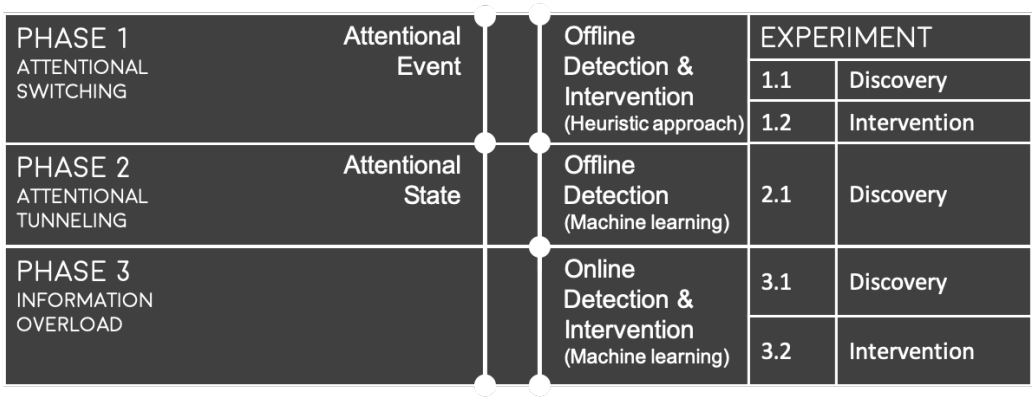


Figure 24. Summary of research approach.

5.1.1. Progression 1: Attentional Events to States

The first goal of this research program was to determine whether PDM can be used to detect attentional processes in participants. To do this, we first needed to determine whether PDM could be used to detect an *attentional event*, which we defined as an attentional process with a clear onset and offset. The first set of experiments focused on *attentional switching*, which is the process of disengaging from an attentional fixation, shifting attention to a new location, and then engaging with a new fixation (Posner & Presti, 1987). This was chosen because of the heavy attentional switch demand within information-dense domains and the findings that these attentional switches incur performance costs (Dombrowe et al., 2011; Longman et al., 2017).

We examined the temporal characteristics of participant interaction to determine if there were predictable points where interruptions occurred. Both Experiments 1.1 and 1.2 identified

significant interruptions in participant interaction prior to moving to a new location in a network graph. Once a participant arrived at that new location there was a second interruption in participant interaction. This interruption pattern aligned with the disengagement and engagement phases of attentional switches, respectively, which validated the use of PDM for detecting attentional events. These findings suggested the viability of using participant interaction data to infer more complex and enduring attentional states.

The second phase of this research advanced to detecting an *attentional state*, which we defined as an enduring attentional process without clearly defined onsets and offsets. The state of interest in Phase 2 was *attentional tunneling*, which we operationalized as the allocation of attention to an information stream for a duration that is longer than optimal given the potential cost of neglecting information outside of that stream. We selected attentional tunneling as the second cognitive bottleneck both because it met the criteria to be considered an attentional state and because of its prevalence within information dense spaces (Rantanen & Goldberg, 1999). Furthermore, its real-world impacts (e.g., Three Mile Island; Rubenstein & Mason, 1979) rendered findings pertaining specifically to attentional tunneling useful by themselves rather than purely serving as a proof of concept for PDM.

Because attentional tunneling is a state rather than an event, detection required machine learning approaches capable of detecting nuanced patterns in user interaction that persisted over a period of time. We employed deep neural networks and were able to successfully detect states of attentional tunneling significantly above chance level. This result suggested that enduring attentional states present themselves in patterns of behaviour and supported advancing to additional attentional states.

In the final set of experiments, we applied machine learning methods to streams of interaction data taken from *information overload* and non-information overload states. Information overload was operationalized as an attentional load surplus precipitated by the excessive presentation of information. Again, information overload represents an attentional state rather than an event and therefore represents a more complex attentional process to detect via PDM. Furthermore, we expected the behavioural characteristics of information overload to be significantly subtler than those of attentional tunneling. Therefore, this phase was intended to serve as a proof of concept for using PDM to classify a wide spectrum of attentional states.

Using PDM in conjunction with relatively simple machine learning methods (i.e., decision trees and random forest classifiers), states of information overload were detected at a rate significantly higher than chance level. Simpler methods than those used to detect attentional tunneling were able to be used as there was significantly more recorded data. These methods were then used in a second experiment to perform online classification of information overload for the purpose of driving an AUI.

The successful completion of this progression supports the viability of using PDM to detect a wide array of attentional processes across a spectrum of domains and tasks.

5.1.2. Progression 2: Offline to Online Detection & Intervention

The second progression from the research plan (see Figure 24) sought to advance PDM from offline to online detection and intervention of maladaptive attentional states. As a proof of concept, the first phase of experimentation employed a heuristic approach to detection and intervention. This phase did not produce a truly adaptive user interface as it assumed that all attentional switches incurred a performance decrement and therefore required an intervention. Experiment 1.1 successfully identified interruptions in participant interaction both before and after an attentional switch. Experiment 1.2 introduced a static intervention strategy that sought to mitigate the performance decrements incurred from attentional switches. A literature review identified the probable connection between attentional switches and *visual momentum,* defined as the process by which information is extracted and integrated when an operator moves to a new point in the display (Woods, 1984). Woods (1984) posited that providing users with context while navigating within an interface will increase visual momentum. Our interventions therefore aimed to contextualize attentional switches to increase the visual momentum within the interface. This intervention reduced the length of the interruption both before and after an attentional switch.

The second phase of this research program sought to implement machine learning techniques that could be used to perform online detection of attentional tunneling. In order to detect attentional tunneling, a more complex machine learning method was required. Convoluted Long Short-Term neural networks (CNN-LSTM) were chosen as these are able to represent long-term temporal dependencies in their internal states and perform classification of time series data (Hochreiter & Schmidhuber, 1997; Pak et al., 2018). These were applied to passively recorded

interaction streams under attentionally tunneled and non-attentionally tunneled states and were able to identify streams of interactions at a rate significantly higher than chance level. Although an intervention experiment was not conducted in Phase 2, verifying the use of these methods supports their viability in online detection of attentional state as their only requirement is appropriately formatted interaction data taken over a period of time. Future studies should explore the efficacy of an attentional tunneling intervention based off of this classifier.

The third phase of this research program tied together the lessons learned over the first two phases by performing online detection and intervention of information overload. This represented a truly adaptive system that monitored user interactions for states of information overload rather than assuming that a singular interaction necessarily resulted in the attentional process of interest, as was the case in Phase 1. Results from this phase showed that the AUIs worked well with users who were more engaged with the experimental task. However, for users who were taking a more economic approach and prioritizing speed over accuracy, the AUIs were significantly worse.

The results from the successful completion of this progression suggest the viability of real-time AUIs driven by PDM.

5.2. Contributions to the field

Past research that has employed PDM (e.g., Mac Aoidh et al., 2012; McDonald et al., 2014; Palmius et al., 2016) has tended to focus on specific cognitive states (e.g., information overload, drowsiness, depression) rather than on the manner by which those states are detected. Although this has produced impressive results with clear applications, it has meant that PDM has yet to be systematically studied. This dissertation addresses this gap by studying PDM across a variety of attentional processes and with a range of methods. It illustrates both the theoretical potential of using PDM to drive adaptive systems as well as the practical utility of these methods for mitigating specific cognitive bottlenecks.

Experiments 1.2 and 3.2 were respectively able to mitigate the negative costs incurred from attentional switches and information overload in some user groups. Although an intervention experiment was not conducted in Phase 2 (i.e., attentional tunneling), the positive results from the completed intervention experiments offers promise for an attentional tunneling intervention.

Furthermore, our findings on attentional tunneling lead directly to design suggestions that can be tested in future research. These results demonstrate that adaptations targeting a specific cognitive bottleneck can improve performance. However, as both past studies and the results from Experiments 3.1 and 3.2 have shown, in information dense spaces, multiple bottlenecks can coexist (e.g., attentional tunneling + information overload; Milgram, 1970). Therefore, future AUIs can monitor for multiple bottlenecks and employ an array of detection and intervention strategies. Furthermore, it is likely that both positive and negative attentional states warrant different design considerations. Future AUIs should also seek to identify positive attentional states and determine what interface adaptations can sustain those states and further facilitate the task (see Limitations and Future Work for additional discussion).

This work also shows that different user groups respond differently to AUIs, which may explain why AUIs have had difficulty gaining widespread adoption. Feigh et al. (2012) state that one of the main hindrances to the effectiveness of AUIs is an insufficient contextual understanding. It seems likely that the openness of a user group to adaptive systems may be a critical aspect of this context. For example, if it is likely that one user group is less open to an AUI (e.g., older users; Lavie & Meyer, 2010), designers may curtail the aggressiveness of adaptations.

Beyond the findings directly relating to AUIs and PDM, this dissertation also produced novel findings on attentional switching, attentional tunneling, and information overload. Phase 1 showed that there is a significant interruption in interaction both before and after an attentional switch, during which people are disengaging from a previous attentional fixation and engaging with a new point, respectively. This finding translated the three phase attentional switch identified by Posner & Presti (1987) to attentional switches that occur beyond someone's visual field. Furthermore, most attentional switch studies employ relatively simple psychological tasks because of the precision required to study attentional switches. Therefore, these results showed that the findings described in the attentional switching literature carry over to more complicated environments. Phase 2 showed that attentional tunnels facilitate performance when information appears within them, but significantly hinder it when information appears outside of them. This finding supported shifting the operationalization of attentional tunnels from the *expected* to the *potential* cost of neglecting information outside of a particular information stream. As expected, Experiment 3.1 demonstrated that information overload significantly increased workload, decreased accuracy, and slowed completion time. Interestingly, however, we found that when

participants were overloaded with information, the time they spent on a task did not significantly affect their accuracy. We also found that participants who were overloaded with information tended to zoom in more, which caused them to take in a smaller portion of a search area. These findings further demonstrate the relationship between information overload and attentional tunneling as well as the performance impacts of this relationship.

This work produced two novel experimental testbeds: ANVEL and CogLog. ANVEL presents a cybersecurity simulation that is particularly useful for machine learning researchers who are interested in examining the relationship between different machine learning recommendation strategies and user behaviour, performance, or situation awareness. A detailed description of ANVEL was published in the 2017 IEEE-SMC conference proceedings (Kortschot et al., 2017). CogLog is more useful for researchers studying PDM. It provides the architecture to develop map-based attentional state induction and to record user interactions while in that state. CogLog_AT.3 was adapted to CogLog_IO.1, which was then adapted to CogLog_IO.2. This adaptation process demonstrates the relative efficiency with which the CogLog platform can be applied to novel attentional states or use cases. Both of these platforms are open source and researchers are encouraged in relevant publications to contact the authors for source code.

Finally, this work produced four large datasets that can be used by future researchers seeking to either study specific aspects of the attentional states that we investigated or to develop and evaluate machine-learning classifiers. One of the benefits stemming from PDM recording all user interactions over the course of an entire experiment is that it allows research questions to be identified after an experiment has been completed. This differs from the majority of behavioural studies where one specific aspect of performance is recorded. For example, in Experiment 1.1, participants' average zoom level tended to increase as they progressed within a trial. This suggests that participants may have been getting increasingly attentionally tunneled as they received more evidence confirming their beliefs about where important information existed within the network. This provides more evidence for the pervasiveness of attentional tunneling and was identified as a potentially interesting research question following the completion of Phase 1. It is likely that there are more novel findings within the data from any of the five experiments that can serve as the basis for future experiments or publications. Furthermore, each of the three phases of this dissertation also produced many hours of valuable time series data that is explicitly labelled and does not contain missing data: Phase 1 generated approximately 40

hours of data on attentional switching; Phase 2 generated approximately 10 hours of data on attentional tunneling; and Phase 3 generated 80 total hours of data on information overload. As machine learning datasets can be both time consuming and expensive to acquire, the datasets generated by this research are a valuable resource for future researchers who are seeking to develop and evaluate novel machine-learning methods. For example, the Human Activity Recognition dataset (Anguita et al., 2013) has been viewed over 800 000 times on the UC Irvine machine learning repository (archive.ics.uci.edu/ml/index.php). This dataset represents a good point of comparison to those generated over the course of this dissertation as it includes various physiological signals under different activities for classification purposes. All publications from this research include contact information for the authors and encourage researchers to inquire about access to the data.

5.3. Limitations and Future Work

The principal limitation of this work was that it was largely proof of concept. This limitation directly influenced a variety of methodological and design decisions such as the use of relatively simple machine learning techniques as well as the limited AUI design. Experiment 2.1 utilized a CNN-LSTM neural network to detect attentional tunneling. This method is widely recognized as being the state of the art for time series classification tasks (Karim, Majumdar, Darabi, & Chen, 2017). However, extracting the CNN-LSTM weights and encoding it using JavaScript would have been difficult. This was not a problem in Phase 2 as no intervention experiment was conducted. However, in Phase 3 where the purpose was to use a PDM inference engine to drive an AUI, this presented an issue. Therefore, Experiment 3.1 employed relatively simple decision trees, which were essentially a series of learned *if-else* rules, which may have compromised the precision of the classifiers and therefore limited their efficacies in Experiment 3.2. Furthermore, as Experiments 1.2 and 3.2 sought to demonstrate the efficacy of targeting a specific bottleneck, the interfaces were designed to isolate that effect rather than to maximize task performance. This may have limited their overall usability and skewed participants' perception of them.

A second significant limitation was that the classifiers were trained on a heterogeneous participant pool. While the use of MTurk allowed large corpuses of data to be generated for Experiments 2.1, 3.1, and 3.2, it also resulted in a diverse sample of participants. This may have resulted in a sample that approached the experimental tasks in fundamentally different ways. As

such, patterns of behaviour that were indicative of an attentional state for one participant may not have translated to a second. In spite of this, the classifiers all performed well above chance and yielded good results in the intervention phases. Future studies should evaluate the efficacy of this approach amongst a more homogenous participant pool. Such participant pools may be more reflective of real-world operating scenarios compared to the participant pools used in this research. Future studies should also seek to develop personalized classifiers that are trained on behaviour from one participant over time. This personalization would accommodate some of the nuances that may separate the behavioural manifestations of attentional states from person to person.

A third significant limitation was that our experimental paradigms never allowed participants to gain significant experience with any of the systems. This is problematic because this research predominantly applied to spaces where real-world users would be experts. The results of Experiment 3.2 demonstrated that participants who were more engaged with the task tended to be more receptive of AUIs. The lack of expertise and possibility of conflicting underlying motivations for completing the experiment may have therefore influenced our results significantly. It is important to consider that using expert participants would not have necessarily ameliorated this problem as experts would be familiar with the interface with which they gained their expertise. Future research should therefore allow participants who are minimally experienced with any particular interface to become experts with the adaptive interfaces such that they are able to develop a persistent mental model of how the interface functions.

Finally, as we were concerned with evaluating the efficacy of PDM as it pertained to the attentional state of interest, we only incorporated one aspect of context in our adaptive designs. As Dorneich et al. (2005) demonstrate, the use of physiological measures can yield adaptive systems that significantly improve performance. Therefore, coupling PDM with physiological state inference has the potential to significantly improve adaptive design. Future studies should augment PDM approaches with system, environmental, task/mission, or spatio/temporal variables to assess the accuracy of the classifier and the benefits yielded from a resulting adaptive interface. Furthermore, future researchers should examine the accuracy of classifiers that combine PDM with physiological signals for performing attentional state inference.

5.4. Conclusions

The finding that behaviour provides insight into cognition was the basis of much of the behaviourism of the 1950's. However, as new methods of directly imaging and sampling physiological signals have developed, the popularity of formal behaviourism has significantly waned (Watrin & Darwich, 2012). This research provides empirical evidence that behaviourism via PDM can provide practical and actionable insights into human cognition and should therefore be considered alongside other contextual cues for drivers of adaptive systems.

References

Aguinis, H., Gottfredson, R. K., & Joo, H. (2013). Best-Practice Recommendations for Defining, Identifying, and Handling Outliers, *16*(2), 270–301. https://doi.org/10.1177/1094428112470848

Ahmad, A., Hadgkiss, J., & Ruighaver, A. B. (2012). Incident response teams - Challenges in supporting the organisational security function. *Computers and Security*, *31*(5), 643–652. https://doi.org/10.1016/j.cose.2012.04.001

Ahn, J.-W., & Brusilovsky, P. (2013). Adaptive visualization for exploratory information retrieval. *Information Processing and Management*, *49*(5), 1139–1164.

Alrajeh, N. A., Khan, S., & Shams, B. (2013). Intrusion Detection Systems in Wireless Sensor Networks: A Review. *International Journal of Distributed Sensor Networks*, *2013*, 1–7. https://doi.org/10.1109/SURV.2013.050113.00191

Angelini, M., & Santucci, G. (2017). Cyber situational awareness: from geographical alerts to high-level management. *Journal of Visualization*, *20*(3), 453–459. https://doi.org/10.1007/s12650-016-0377-3

Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A Public Domain Dataset for Human Activity Recognition Using Smartphones. In *21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Bruges, Belgium.

Arif, M., & Kattan, A. (2015). Physical activities monitoring using wearable acceleration sensors attached to the body. *PLoS ONE*, *10*(7), 1–16. https://doi.org/10.1371/journal.pone.0130851

Ashfaq, A. B., & Khayam, S. A. (2011). Evaluation of Contemporary Anomaly Detection Systems (ADSs). In J. A. Zubairi & A. Mahboob (Eds.), *Cyber Security Standards, Practices and …* (pp. 90–112). Hershey, PA: Information Science Reference. https://doi.org/10.4018/978-1-60960-851-4.ch006

Atchley, P., & Dressel, J. (2004). Conversation Limits the Functional Field of View. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *46*(4), 664–673. https://doi.org/10.1518/hfes.46.4.664.56808

Bainbridge, L. (1983). Ironies of Automation. *Automatica*, *19*(6), 775–779.

Balakrishnan, G., Durand, F., & Guttag, J. (2013). Detecting pulse from head motions in video. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 3430–3437). https://doi.org/10.1109/CVPR.2013.440

Baldwin, C. L., Roberts, D. M., Barragan, D., Lee, J. D., Lerner, N., & Higgins, J. S. (2017). Detecting and Quantifying Mind Wandering during Simulated Driving. *Frontiers in Human Neuroscience*, *11*(August), 1–15. https://doi.org/10.3389/fnhum.2017.00406

Ben-Asher, N., & Gonzalez, C. (2015). Effects of cyber security knowledge on attack detection. *Computers in Human Behavior*, *48*, 51–61. https://doi.org/10.1016/j.chb.2015.01.039

Bennett, K. B., Bryant, A., & Sushereba, C. (2018). Ecological Interface Design for Computer Network Defense. *Human Factors*, *60*(5), 610–625. https://doi.org/10.1177/0018720818769233

Bennett, K. B., & Flach, J. M. (2012). Visual momentum redux. *International Journal of Human Computer Studies*, *70*(6), 399–414. https://doi.org/10.1016/j.ijhcs.2012.01.003

Benselin, J. C., & Ragsdell, G. (2016). Information overload: The differences that age makes. *Journal of Librarianship and Information Science*, *48*(3), 284–297. https://doi.org/10.1177/0961000614566341

Benyon, D., & Murray, D. (1993). Applying user modeling to human-computer interaction design. *Artificial Intelligence Review*, *7*(3–4), 199–225. https://doi.org/10.1007/BF00849555

Berka, C., Levendowski, D. J., Lumicao, M. N., Yau, A., Davis, G., Zivkovic, V. T., … Craven, P. L. (2007). EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks. *Aviation, Space, and Environmental Medicine*, *78*(5 Suppl), B231-44. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/17547324

Besner, D., Reynolds, M., & O'Malley, S. (2009). When underadditivity of factor effects in the Psychological Refractory Period paradigm implies a bottleneck: Evidence from psycholinguistics. *The Quarterly Journal of Experimental Psychology*, *62*(11), 2222–2234. https://doi.org/10.1080/17470210902747187

Borghetti, B. J., Giametta, J. J., & Rusnock, C. F. (2017). Assessing Continuous Operator Workload with a Hybrid Scaffolded Neuroergonomic Modeling Approach. *Human Factors*, *59*(1), 134–146. https://doi.org/10.1177/0018720816672308

Borst, J. P., Taatgen, N. A., & van Rijn, H. (2010). The problem state: A cognitive bottleneck in multitasking. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *36*(2), 363–382. https://doi.org/10.1037/a0018106

Briggs, G. F., Hole, G. J., & Turner, J. A. J. (2018). The impact of attentional set and situation awareness on dual tasking driving performance. *Transportation Research Part F: Traffic Psychology and Behaviour*, *57*, 36–47. https://doi.org/10.1016/j.trf.2017.08.007

Brooke, J. (1996). SUS: A "quick and dirty" usability scale. In *Usability Evaluation in Industry*. London: Taylor and Francis.

Budd, A., & Björklund, E. (2016). Cutting-edge Visual Effects. In *CSS Mastery* (pp. 335–370). Berkeley, CA: Apress.

Byers, J. C., Bittner, A. C., & Hill, S. G. (1989). Traditional and Raw Task Load Index (TLX) correlations: are paired comparisons necessary? In A. Mital & B. Das (Eds.), *Advances in Industrial Ergonomics and Safety* (vol. 1., pp. 481–488). London: Taylor & Francis.

Carroll, L. N., Au, A. P., Detwiler, L. T., Fu, T. chieh, Painter, I. S., & Abernethy, N. F. (2014). Visualization and analytics tools for infectious disease epidemiology: A systematic review. *Journal of Biomedical Informatics*, *51*, 287–298. https://doi.org/10.1016/j.jbi.2014.04.006

Chan, T. C. Y., Cho, J. A., & Novati, D. C. (2012). Quantifying the contribution of NHL player types to team performance. *Interfaces*, *42*(2), 131–145. https://doi.org/10.1287/inte.1110.0612

Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Neurocomputing Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing*, *307*, 72–77. https://doi.org/10.1016/j.neucom.2018.03.067

Cockburn, A., Karlson, A., & Bederson, B. B. (2009). A review of overview+detail, zooming, and focus+context interfaces. *ACM Computing Surveys*, *41*(1), 1–31. https://doi.org/10.1145/1456650.1456652

Corchado, E., & Herrero, Á. (2011). Neural visualization of network traffic data for intrusion detection. *Applied Soft Computing Journal*, *11*(2), 2042–2056. https://doi.org/10.1016/j.asoc.2010.07.002

D'Amico, A., Whitley, K., Tesone, D., O'Brien, B., & Roth, E. M. (2005). Achieving Cyber Defense Situational Awareness: A Cognitive Task Analysis of Information Assurance Analysts. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *49*, 229–233. https://doi.org/10.1177/154193120504900304

Dadashi, N., Golightly, D., & Sharples, S. (2017). Seeing the woods for the trees: the problem of information inefficiency and information overload on operator performance. *Cognition, Technology & Work*, *19*(4), 561–570. https://doi.org/10.1007/s10111-017-0451-1

Dale, R. (2016). The Return of Chatbots. *Natural Language Engineering*, *22*(5), 811–817. https://doi.org/10.1017/S1351324916000243

Dixon, B. J., Daly, M. J., Chan, H., Vescan, A. D., Witterick, I. J., & Irish, J. C. (2013). Surgeons blinded by enhanced navigation : the effect of augmented reality on attention, 454–461. https://doi.org/10.1007/s00464-012-2457-3

Dombrowe, I., Donk, M., & Olivers, C. N. L. (2011). The costs of switching attentional sets. *Attention, Perception, and Psychophysics*, *73*(8), 2481–2488. https://doi.org/10.3758/s13414-011-0198-3

Donmez, B., Boyle, L. N., & Lee, J. D. (2006). The Impact of Distraction Mitigation Strategies on Driving Performance. *Human Factors*, *48*(4), 785–804.

Dorneich, M. C., Mathan, S., Ververs, P. M., & Whitlow, S. D. (2007). An Evaluation of Real-Time Cognitive State Classification in a Harsh Operational Environment. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, *51*, 146–150. https://doi.org/10.1177/154193120705100401

Dorneich, M. C., Whitlow, S. D., Mathan, S., Carciofini, J., & Ververs, P. M. (2005). The communications scheduler: A task scheduling mitigation for a closed loop adaptive system. In D. D. Schmorrow (Ed.), *Foundations of Augmented Cognition* (pp. 132–141). Mahwah, N.J.: Lawrence Erlbaum Associates.

Dorneich, M. C., Whitlow, S. D., Mathan, S., May Ververs, P., Erdogmus, D., Adami, A., … Lan, T. (2007). Supporting Real-Time Cognitive State Classification on a Mobile Individual. *Journal of Cognitive Engineering and Decision Making*, *1*(3), 240–270. https://doi.org/10.1518/155534307X255618

Dorneich, M. C., Whitlow, S. D., Ververs, P. M., & Rogers, W. H. (2003). Mitigating cognitive bottlenecks via an augmented cognition adaptive system. *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483)*, *1*, 937–944 vol.1. https://doi.org/10.1109/ICSMC.2003.1243935

Duncan, J. (1980). The locus of interference in the perception of simultaneous stimuli. *Psychological Review*, *87*(3), 272–300. https://doi.org/10.1037/0033-295X.87.3.272

Enstrom, K. D., & Rouse, W. B. (1977). Real-Time Determination of How a Human Has Allocated His Attention Between Control and Monitoring Tasks. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-7*(3), 153–161. https://doi.org/10.1109/TSMC.1977.4309679

Eppler, M. J., & Mengis, J. (2004). The concept of information overload: A review of literature from organization science, accounting, marketing, MIS, and related disciplines. *Information Society*, *20*(5), 325–344. https://doi.org/10.1080/01972240490507974

Ettwig, J. F., & Bronkhorst, A. W. (2015). Attentional switches and dual-task interference. *PLoS ONE*, *10*(3). https://doi.org/10.1371/journal.pone.0118216

Feigh, K. M., Dorneich, M. C., & Hayes, C. C. (2012). Toward a Characterization of Adaptive Systems: A Framework for Researchers and System Designers. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *54*(6), 1008–1024. https://doi.org/10.1177/0018720812443983

Fisher, C. W., & Kingma, B. R. (2001). Criticality of data quality as exemplified in two disasters. *Information and Management*, *39*(2), 109–116. https://doi.org/10.1016/S0378-7206(01)00083-0

Forsman, P. M., Vila, B. J., Short, R. A., Mott, C. G., & Van Dongen, H. P. A. (2013). Efficient driver drowsiness detection at moderate levels of drowsiness. *Accident Analysis and Prevention*, *50*, 341–350. https://doi.org/10.1016/j.aap.2012.05.005

Franke, U., & Brynielsson, J. (2014). Cyber situational awareness – a systematic review of the literature. *Computers & Security*, *46*, 41. https://doi.org/10.1016/j.cose.2014.06.008

Friedman, R. S., Fishbach, A., Förster, J., & Werth, L. (2003). Attentional Priming Effects on Creativity. *Creativity Research Journal*, *15*(2–3), 277–286. https://doi.org/10.1080/10400419.2003.9651420

Furnas, G. W. (1986). Generalized fisheye views. In *ACM SIGCHI Bulletin* (pp. 16–23). https://doi.org/10.1145/22339.22342

Gomez-Uribe, C. A., & Hunt, N. (2015). The Netflix Recommender System : Algorithms , Business Value, and Innovation. *ACM Transactions on Management Information Systems (TMIS)*, *6*(4), 1–19.

Goodall, J. R., Lutters, W. G., & Komlodi, A. (2009). Developing expertise for network intrusion detection. *Information Technology & People*, *22*(2), 92–108. https://doi.org/10.1108/09593840910962186

Gorgoglione, M., Panniello, U., & Tuzhilin, A. (2019). Recommendation strategies in personalization applications. *Information and Management*, (January), 1–12. https://doi.org/10.1016/j.im.2019.01.005

Goutam, R. K. (2015). Importance of Cyber Security. *International Journal of Computer Applications*, *111*(7), 14–18.

Grawemeyer, B., & Cox, R. (2005). A Bayesian Approach to Modelling Users' Information Display Preferences. In *International Conference on User Modeling* (pp. 225–230). https://doi.org/10.1007/11527886_29

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2017). LSTM:

A Search Space Odyssey. *Transactions on Neural Networks and Learning Systems*, *28*(10), 2222–2232. Retrieved from http://arxiv.org/abs/1503.04069

Gutzwiller, R., Ferguson-Walter, K., Fugate, S., & Rogers, A. (2018). “Oh, look, a butterfly!” A framework for distracting attackers to improve cyber defense. In *Proceedings of the Human Factors and Ergonomics Society* (pp. 272–276). Philadelphia. https://doi.org/10.1177/1541931218621063

Haase, R. F., Jome, L. M., Ferreira, J. A., Santos, E. J., Connacher, C. C., & Sendrowitz, K. (2014). Individual Differences in Capacity for Tolerating Information Overload Are Related to Differences in Culture and Temperament Individual Differences in Capacity for Tolerating Information Overload Are Related. *Journal of Cross-Cultural Psychology*, (August 2016). https://doi.org/10.1177/0022022113519852

Hall, A., & Walton, G. (2004). Information overload within the health care system: a literature review. *Health Information and Libraries Journal*, *21*(2), 102–108. https://doi.org/10.1111/j.1471-1842.2004.00506.x

Hart, S. G., & Staveland, L. E. (1988). Developent of a multi-dimensional workload rating scale: Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Human Mental Workload* (pp. 139–183). Amsterdam, The Netherlands: Elsevier.

Hauser, D. J., & Schwarz, N. (2016). Attentive Turkers: MTurk participants perform better on online attention checks than do subject pool participants. *Behavior Research Methods*, *48*(1), 400–407. https://doi.org/10.3758/s13428-015-0578-z

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hodgetts, H. M., Vachon, F., Chamberland, C., & Tremblay, S. (2017). See No Evil: Cognitive Challenges of Security Surveillance and Monitoring. *Journal of Applied Research in Memory and Cognition*, *6*(3), 230–243. https://doi.org/10.1016/j.jarmac.2017.05.001

Höök, K. (2000). Steps to take before intelligent user interfaces become real. *Interacting with Computers*, *12*(4), 409–426. https://doi.org/doi:10.1016/S0953-5438(99)00006-5

Hopf, J.-M., Boehler, C. N., Schoenfeld, M. A., Heinze, H.-J., & Tsotsos, J. K. (2010). The spatial profile of the focus of attention in visual search: Insights from MEG recordings. *Vision Research*, *50*(14), 1312–1320. https://doi.org/10.1016/j.visres.2010.01.015

Horvitz, E., Kadie, C., Paek, T., & Hovel, D. (2003). Models of attention in computing and communication. *Communications of the ACM*, *46*(3), 52. https://doi.org/10.1145/636772.636798

Hosub Lee, Young Sang Choi, & Yeo-Jin Kim. (2011). An adaptive user interface based on spatiotemporal structure learning. *IEEE Communications Magazine*, *49*(6), 118–124. https://doi.org/10.1109/MCOM.2011.5783996

Ishihara, S. (1937). *The series fo plates designed as tests for color blindness*. Tokyo: Kanehara & Co.

Janczyk, M., & Kunde, W. (2010). Does dorsal processing require central capacity ? More evidence from the PRP paradigm, 89–100. https://doi.org/10.1007/s00221-010-2211-9

Karim, F., Majumdar, S., Darabi, H., & Chen, S. (2017). LSTM Fully Convolutional Networks

for Time Series Classification. *IEEE Access*, 1–7. https://doi.org/10.1109/ACCESS.2017.2779939

Karim, F., Majumdar, S., Darabi, H., & Harford, S. (2018). Multivariate LSTM-FCNs for Time Series Classification. *IEEE Access*, 1–9. Retrieved from http://arxiv.org/abs/1801.04503

Katidioti, I., & Taatgen, N. A. (2014). Choice in multitasking: How delays in the primary task turn a rational into an irrational multitasker. *Human Factors*, *56*(4), 728–736.

Khan, S., Gani, A., Abdul Wahab, A. W., Shiraz, M., & Khan, I. A. (2016). Network forensics: Review, taxonomy, and open challenges. *Journal of Network and Computer Applications*, *66*, 214–235. https://doi.org/10.1016/j.jnca.2016.03.005

Kortschot, S. W., & Jamieson, G. A. (2019). Classification of attentional tunneling through behavioral indices. *Human Factors*. https://doi.org/10.1177/0018720819857266

Kortschot, S. W., Jamieson, G. A., & Prasad, A. (in review). Detecting and Treating Information Overload with an Adaptive User Interface. *Human Factors*.

Kortschot, S. W., Sovilj, D., Jamieson, G. A., Sanner, S., Carrasco, C., & Soh, H. (2018). Measuring and Mitigating the Costs of Attentional Switches in Active Network Monitoring for Cybersecurity. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *60*(7), 962–977. https://doi.org/10.1177/0018720818784107

Kortschot, S. W., Sovilj, D., Soh, H., Jamieson, G. A., Sanner, S., Carrasco, C., … Ralph, S. (2017). An open source adaptive user interface for network monitoring. In *IEEE International Conference on Systems, Man and Cybernetics*. Banff.

Kramer, A. F. (1991). Physiological metrics of mental workload: A review of recent progress. In D. L. Damos (Ed.), *Multiple task performance* (pp. 279–328). Bristol, PA: Taylor & Francis.

Lance, G. N., & Williams, W. T. (1966). A generalized sorting strategy for computing classification. *Nature*, *212*, 218.

Lavie, T., & Meyer, J. (2010). Benefits and costs of adaptive user interfaces. *International Journal of Human Computer Studies*, *68*(8), 508–524. https://doi.org/10.1016/j.ijhcs.2010.01.004

Lee, E. A. (2008). Cyber physical systems: Design challenges. *Proceedings - 11th IEEE Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, ISORC 2008*, 363–369. https://doi.org/10.1109/ISORC.2008.25

Lee, Y. Y., & Hsieh, S. (2014). Classifying different emotional states by means of EEG-based functional connectivity patterns. *PLoS ONE*, *9*(4). https://doi.org/10.1371/journal.pone.0095415

Li, P., Salour, M., & Su, X. (2008). A survey of internet worm detection and containment. *IEEE Communications Surveys and Tutorials*, *10*(1), 20–35. https://doi.org/10.1109/COMST.2008.4483668

Li, W., Mo, W., Zhang, X., Squiers, J. J., Lu, Y., Sellke, E. W., … Thatcher, J. E. (2015). Outlier detection and removal improves accuracy of machine learning approach to multispectral burn diagnostic imaging. *Journal of Biomedical Optics*, *20*(12), 1–9. https://doi.org/10.1117/1.JBO.20.12.121305

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A Critical Review of Recurrent Neural Networks for Sequence Learning, 1–38. https://doi.org/10.1145/2647868.2654889

Lipton, Z. C., Kale, D. C., Elkan, C., & Wetzel, R. (2016). Learning to Diagnose with LSTM Recurrent Neural Networks. In *ICLR* (pp. 1–18). San Juan, Puerto Rico. https://doi.org/10.14722/ndss.2015.23268

Longman, C. S., Lavric, A., & Monsell, S. (2017). Self-paced preparation for a task switch eliminates attentional inertia but not the performance switch cost. *Journal of Experimental Psychology: Learning Memory and Cognition*, *43*(6), 862–873. https://doi.org/10.1037/xlm0000347

Lopes, C. T., Franz, M., Kazi, F., Donaldson, S. L., Morris, Q., & Bader, G. D. (2010). Cytoscape Web : an interactive web-based network browser. *Bioinformatics*, *26*(18), 2347–2348. https://doi.org/10.1093/bioinformatics/btq430

Mac Aoidh, E., Bertolotto, M., & Wilson, D. C. (2012). Towards dynamic behavior-based profiling for reducing spatial information overload in map browsing activity. *Geoinformatica*, *16*, 409–434. https://doi.org/10.1007/s10707-011-0137-4

Mangos, P. M., & Hulse, N. A. (2017). Advances in machine learning applications for scenario intelligence: deep learning. *Theoretical Issues in Ergonomics Science*, *18*(2), 184–198. https://doi.org/10.1080/1463922X.2016.1166406

Mannaru, P., Balasingam, B., Pattipati, K., Sibley, C., & Coyne, J. (2016). Cognitive context detection for adaptive automation. *Proceedings of the Human Factors and Ergonomics Society*, 223–227. https://doi.org/10.1177/1541931213601050

Marksteiner, P. R. (2009). Data smog, techno creep and the hobbling of the cognitive dimension. In K. J. Knapp (Ed.), *Cyber Security and Global Information Assurance: Threat Analysis and Response Solutions* (pp. 141–163). Hershey, PA: Information Science Reference. https://doi.org/10.4018/978-1-60566-326-5.ch007

Mason, S. J., & Graham, N. E. (2002). Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. *Quarterly Journal of the Royal Meteorological Society*, *128*(584 PART B), 2145–2166. https://doi.org/10.1256/003590002320603584

McDonald, A. D., Lee, J. D., Schwarz, C., & Brown, T. L. (2014). Steering in a random forest: Ensemble learning for detecting drowsiness-related lane departures. *Human Factors*, *56*(5), 986–998. https://doi.org/10.1177/0018720813515272

McLane, H. C., Berkowitz, A. L., Patenaude, B. N., Mckenzie, E. D., Wolper, E., Wahlster, S., … Mateen, F. J. (2015). Availability, accessibility, and affordability of neurodiagnostic tests in 37 countries. *Neurology*, *85*(18), 1614–1622. https://doi.org/10.1212/WNL.0000000000002090

Michel, M., Helmick, N., & Mayron, L. (2011). Cognitive cyber situational awareness using virtual worlds. In *2011 IEEE International Mutli-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, CogSIMA* (pp. 179–182).

Milgram, S. (1970). The experience of living in cities. *Science*, *167*(3924), 1461–1468.

Mills, K. C., Spruill, S. E., Kanne, R. W., Parkman, K. M., & Zhang, Y. (2001). The Influence of Stimulants, Sedatives, and Fatigue on Tunnel Vision: Risk Factors for Driving and Piloting.

Human Factors: The Journal of the Human Factors and Ergonomics Society, *43*(2), 310–327. https://doi.org/10.1518/001872001775900878

Misra, S., & Stokols, D. (2012). Psychological and Health Outcomes of Perceived Information Overload. *Environment and Behavior*, *44*(6), 737–759. https://doi.org/10.1177/0013916511404408

Mitropoulos, S., Patsos, D., & Douligeris, C. (2006). On Incident Handling and Response: A state-of-the-art approach. *Computers and Security*, *25*(5), 351–370. https://doi.org/10.1016/j.cose.2005.09.006

Moar, J. (2015). *The Future of Cybercrime & Security: Financial and Corporate Threats & Mitigation*. Hampshire, UK. Retrieved from https://www.juniperresearch.com/press/press-releases/cybercrime-cost-businesses-over-2trillion

Morales-Herrera, R., Fernández-Caballero, A., Somolinos, J. A., & Sira-Ramírez, H. (2017). Integration of Sensors in Control and Automation Systems. *Journal of Sensors*, *2017*, 1–2. https://doi.org/10.1155/2017/6415876

Mukhopadhyay, S. C., & Lay-Ekuakille, A. (Eds.). (2010). *Advances in biomedical sensing, measurements, instrumentaiton, and systems*. Berlin: Springer Verlag.

Nagare, R., Plitnick, B., & Figueiro, M. G. (2019). Does the iPad Night Shift mode reduce melatonin suppression? *Lighting Research and Technology*, *51*(3), 373–383. https://doi.org/10.1177/1477153517748189

Nobre, A. C., & Kastner, S. (Eds.). (2014). *The Oxford Handbook of Attention*. Oxford, UK: Oxford University Press.

Nowak, S., & Rüger, S. (2010). How Reliable are Annotations via Crowdsourcing ? In *MIR* (pp. 557–566). Philadelphia.

Olszewski, D. (2014). Fraud detection using self-organizing map visualizing the user profiles. *Knowledge-Based Systems*, *70*, 324–334. https://doi.org/10.1016/j.knosys.2014.07.008

Pak, U., Kim, C., Ryu, U., Sok, K., & Pak, S. (2018). A hybrid model based on convolutional neural networks and long short-term memory for ozone concentration prediction. *Air Quality, Atmosphere & Health*. https://doi.org/https://doi.org/10.1007/s11869-018-0585-1 A

Palmius, N., Tsanas, A., Saunders, K. E. A., Bilderbeck, A. C., Geddes, J. R., Goodwin, G. M., & De Vos, M. (2016). Detecting bipolar depression from Geographic location data. *IEEE Transactions on Biomedical Engineering*, 1–12. https://doi.org/10.1109/TBME.2016.2611862

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, *12*, 2825–2830. https://doi.org/10.1007/s13398-014-0173-7.2

Pfleeger, S. L., & Caputo, D. D. (2012). Leveraging behavioral science to mitigate cyber security risk. *Computers & Security*, *31*, 597–611. https://doi.org/10.1016/j.cose.2011.12.010

Posner, M. I., & Presti, D. E. (1987). Selective attention and cognitive control. *Trends in Neurosciences*, *10*(1), 13–17. https://doi.org/10.1016/0166-2236(87)90116-0

Prieto, O. J., Alonso-González, C. J., & Rodríguez, J. J. (2015). Stacking for multivariate time series classification. *Pattern Analysis and Applications*, *18*(2), 297–312.

https://doi.org/10.1007/s10044-013-0351-9

Rantanen, E. M., & Goldberg, J. H. (1999). The effect of mental workload on the visual field size and shape. *Ergonomics*, *42*(6), 816–834. https://doi.org/10.1080/001401399185315

Régis, N., Dehais, F., Rachelson, E., Thooris, C., Pizziol, S., Causse, M., & Tessier, C. (2014). Formal detection of attentional tunneling in human operator-automation interactions. *IEEE Transactions on Human-Machine Systems*, *44*(3), 326–336. https://doi.org/10.1109/THMS.2014.2307258

Rogé, J., Kielbasa, L., & Muzet, A. (2002). Deformation of the useful visual field with state of vigilance, task priority, and central task complexity. *Perceptual and Motor Skills*, *95*(1), 118–130. https://doi.org/10.2466/pms.2002.95.1.118

Rothrock, L., Koubek, R., Fuchs, F., Haas, M., & Salvendy, G. (2002). Review and reappraisal of adaptive interfaces: toward biologically inspired paradigms. *Theoretical Issues in Ergonomics Science*, *3*(1), 47–84.

Rubenstein, E., & Mason, J. F. (1979). An analysis of Three Mile Island: The accident that shouldn't have happened. *IEEE Spectrum*, *13*(9), 33–42.

Salvucci, D. D., & Taatgen, N. A. (2008). Threaded Cognition: An Integrated Theory of Concurrent Multitasking. *Psychological Review*, *115*(1), 101–130. https://doi.org/10.1037/0033-295X.115.1.101

Sarkar, M., & Brown, M. H. (1992). Graphical fisheye views of graphs. In *Proceedings of the 1992 SIGCHI conference on Human Factors in Computing Systems (CHI)* (pp. 83–91). Monterey, California, USA. https://doi.org/10.1145/142750.142763

Shappell, S. A., & Wiegmann, D. A. (2003). *A Human Error Analysis of General Aviation Controlled Flight Into Terrain Accidents Occurring Between 1990-1998*. *Final Report*. Washington, DC. Retrieved from papers2://publication/uuid/7B74962E-8A03-419E-8C35-019D58D279A0

Sohn, M.-H., & Anderson, J. R. (2001). Task Preparation and Task Repetition: Two-Component Model of Task Switching. *Journal of Experimental Psychology: General*, *130*(4), 764–778. https://doi.org/10.1037/0096-3445.130.4.764

Strauch, B. (2018). Ironies of Automation: Still Unresolved after All These Years. *IEEE Transactions on Human-Machine Systems*, *48*(5), 419–433. https://doi.org/10.1109/THMS.2017.2732506

Swain, M., & Haka, S. (2000). Effects of Information Load on Capital Budgeting Decisions. *Behavioral Research in Accounting*, *12*(2), 171–198. Retrieved from http://search.proquest.com.proxy1.athensams.net/docview/203299454/fulltextPDF/3A0739A6C09B4BC1PQ/9?accountid=14483

Tan, J. H., Hagiwara, Y., Pang, W., Lim, I., Oh, S. L., Adam, M., … Acharya, U. R. (2018). Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals. *Computers in Biology and Medicine*, *94*, 19–26. https://doi.org/10.1016/j.compbiomed.2017.12.023

Tyworth, M., Giacobe, N. A., & Mancuso, V. (2012). Cyber situation awareness as distributed socio-cognitive work. *Cyber Sensing 2012*, *8408*(Level 2), 1–9. https://doi.org/10.1117/12.919338

Valenza, G., Citi, L., Gentili, C., Lanatá, A., Scilingo, E. P., & Barbieri, R. (2015). Characterization of depressive states in bipolar patients using wearable textile technology and instantaneous heart rate variability assessment. *IEEE Journal of Biomedical and Health Informatics*, *19*(1), 263–274. https://doi.org/10.1109/JBHI.2014.2307584

van Dongen, K., & van Maanen, P.-P. (2013). A framework for explaining reliance on decision aids. *Journal of Human Computer Studies*, *71*(4), 410–424. https://doi.org/10.1016/j.ijhcs.2012.10.018

Verwey, W. B., Shea, C. H., & Wright, D. L. (2015). A cognitive framework for explaining serial processing and sequence execution strategies. *Psychon Bull Rev*, *22*, 54–77. https://doi.org/10.3758/s13423-014-0773-4

Virvou, M. (1999). Automatic reasoning and help about human errors in using an operating system. *Interacting with Computers*, *11*(5), 545–573. https://doi.org/10.1016/S0953-5438(98)00043-5

Vrigkas, M., Nikou, C., & Kakadiaris, I. A. (2015). A Review of Human Activity Recognition Methods. *Frontiers in Robotics and AI*, *2*(November), 1–28. https://doi.org/10.3389/frobt.2015.00028

Watrin, J. P., & Darwich, R. (2012). On behaviorism in the cognitive revolution: Myth and reactions. *Review of General Psychology*, *16*(3), 269–282. https://doi.org/10.1037/a0026766

Werlinger, R., Muldner, K., Hawkey, K., & Beznosov, K. (2010). Preparation, detection, and analysis: the diagnostic work of IT security incident response. *Information Management & Computer Security*, *18*(1), 26–42. https://doi.org/10.1108/09685221011035241

Wickens, C. D. (2008). Multiple Resources and Mental Workload. *Human Factors*, *50*(3), 449–455. https://doi.org/10.1518/001872008X288394.

Wickens, C. D., & Alexander, A. L. (2009). Attentional Tunneling and Task Management in Synthetic Vision Displays. *The International Journal of Aviation Psychology*, *19*(2), 182–199. https://doi.org/10.1080/10508410902766549

Wickens, C. D., & Gutzwiller, R. S. (2017). The Status of the Strategic Task Overload Model (STOM) for Predicting Multi-Task Management. In *Proceedings of the Human Factors and Ergonomics Society* (pp. 757–761). Austin. https://doi.org/10.1177/1541931213601674

Wickens, C. D., Gutzwiller, R. S., & Santamaria, A. (2015). Discrete task switching in overload : A meta-analyses and a model. *Journal of Human Computer Studies*, *79*, 79–84. https://doi.org/10.1016/j.ijhcs.2015.01.002

Williams, L. J. (1995). Visual field narrowing induced by workload. *The Journal of General Psychology*, *122*(2), 225–235.

Wilson, G. F., & Russell, C. A. (2004). Operator Functional State Classification Using Multiple Psychophysiological Features in an Air Traffic Control Task. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, *45*(3), 381–389. https://doi.org/10.1518/hfes.45.3.381.27252

Woods, D. D. (1984). Visual momentum: a concept to improve the cognitive coupling of person and computer. *Int. J. Man-Machine Studies*, *21*, 229–244. Retrieved from http://ac.els-cdn.com/S0020737384800437/1-s2.0-S0020737384800437-main.pdf?_tid=a6312d00-238a-11e7-9c20-00000aacb360&acdnat=1492446585_63d6b2c8437eedd810baaa400937658b

Woods, D. D., & Patterson, E. S. (2000). How Unexpected Events Produce An Escalation Of Cognitive And Coordinative Demands. In P. A. Hancock & P. A. Desmond (Eds.), *Stress Workload and Fatigue* (pp. 1–15). Hillsdale, N.J.: Lawrence Erlbaum Associates. https://doi.org/10.1097/ACM.0b013e3181ea3831

Wu, L., Zhu, Z., Cao, H., & Li, B. (2016). Influence of information overload on operator's user experience of human–machine interface in LED manufacturing systems. *Cognition, Technology and Work*, *18*(1), 161–173. https://doi.org/10.1007/s10111-015-0352-0

Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., & Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, *11*(2), 68–75. https://doi.org/10.1049/iet-its.2016.0208

Appendix A - Literature Review

5.5. Attentional States/Events

This dissertation focused on two classes of psychological phenomena: Attentional events, and states. We broadly define an attentional event as a psychological phenomenon with a clear onset or offset. We define an attentional state as an enduring psychological phenomenon that can persist for a variable period of time. The below sections will present additional research on methods used to assess the attentional phenomena studied in this dissertation.

5.5.1. Attentional Event – Attentional Switching

The first chapter of this dissertation examined attentional switches, which Posner & Presti (1987) define as a three phase process consisting of a disengagement from a previous fixation, shift to a new location, and re-engagement with the new attentional set.

The main method for studying attentional switches is through eye tracking. For example, Longman, Lavric, and Monsell (2013) employed eye-tracking methods to study the relationship between performance and reorientation during attentional switches. Foerster and Schneider (2015) used eye-tracking to study the effect of long term memory on attentional switches. One of the main reasons that eye-tracking is well-suited to the study of attentional switches is because of the precision of measurement required.

The majority of attentional switching literature focuses on overt attentional switches, which occur when an attentional switch corresponds to physical eye movement (McCoy & Theeuwes, 2018). However, there are also covert attentional switches, which are defined as an attentional switch with no associated shift of gaze (Posner & Presti, 1987). Although these represent a distinct class of attentional switches, it should be noted that overt switches are necessarily preceded by a covert switch (Foerster & Schneider, 2015). Covert attention can be applied to a variety of perceptual tasks, including object tracking (Baurès, Bennett, & Causer, 2014), target detection and identification (Goolkasian & Tarantino, 1999). Interestingly, even covert

attentional switches have been shown to manifest through eye movement (Wedel, Pieters, & Liechty, 2003), which raises the question of whether they can also be studied through PDM.

5.5.2. Attentional State – Attentional Tunneling

We modified the Wickens and Alexander (2009) definition of attentional tunneling, describing it as the excessive allocation of attentional to a particular information stream given the *potential* rather than the *expected* cost of neglecting information outside of that stream. The literature review included in Chapter 3 focused on the practical outcomes of attentional tunneling. However, the majority of psychological research in this field has focused on the cognitive causes and effects of this phenomenon.

For example, Freeman, Macaluso, Rees, and Driver (2014) examined neurological correlates of focused attention and found that task-irrelevant stimuli were suppressed in relation to how strongly they competed with the task-relevant stimuli. This partially explains the manner by which attentional tunneling can be conflated with focused attention. A second example comes from Milham, Banich, Claus, and Cohen (2003), who demonstrated that the dorsolateral prefrontal cortex was key in imposing top-down attentional control, which suggests that it was likely key in driving the attentional tunneling that we observed in Experiment 2.1. Interestingly, they found that practice reduced the amount of activity required to process irrelevant stimuli, a finding that is also likely related to our attentional tunneling findings.

5.5.3. Attentional State – Information Overload

Similar to attentional tunneling, the literature review included in Chapter 4 centred around the practical outcomes of information overload. However, a significant amount of research has focused on the neurological representation of information overload.

For example, Jaeggi et al. (2007) demonstrated a strong relationship between neurological response to information overload and performance level. They were also able to localize these differences to predominantly the prefrontal and posterior areas. Matsuyoshi, Osaka, and Osaka (2014) showed significant age-related differences in terms of information overload threshold and the neuronal representation of that overload.

5.6. Behaviour-Based Inference & Intervention

Although PDM is becoming increasingly common, the number of instances where it is actually used to trigger an adaptation are relatively sparse. For example, Palmius et al. (2014) were able to successfully detect impending state of depression, but did not introduce an intervention aimed at mitigating the severity of that episode. Similarly, McDonald et al. (2014) detected states of drowsiness in drivers, but also did not introduce an intervention aimed at either increasing their arousal or informing them to pull over.

A significant class of behaviorally-driven interventions comes in the form of personalized recommender systems, wherein collaborative filtering approaches are applied to users to determine future recommendations (Gomez-Uribe & Hunt, 2015). This approach could be considered as falling under the PDM methodology as the algorithms used to deliver recommendations are based on data that is inherent to the task. There are notable differences between this approach however and the research described in this dissertation. First, these systems are not responding to an internal psychological phenomenon. Rather, they identify users who have demonstrated similar behavioural characteristics and recommend future items based on the behaviour of these other users. An example of an AUI driven by PDM within this context would be to adapt the recommendations based on a real-time inferred attentional state of the user. For example, there may be characteristics in how a user browses recommendations (e.g., scroll speed, cursor dynamism, etc.) that suggest their current attentional capacity. Based off of these inferences you may modify the recommendations. The second important distinction is that the adaptations occur at the loading phase rather than dynamically as the user interacts with the interface.

The closest working example that we were able to find in the literature of a behaviourally-based adaptive intervention comes from Okoshi et al. (2016), who used user interaction data to identify when users were switching between tasks. They then implemented a system that only pushed notifications during those periods so as not to interrupt any individual task. Their system significantly reduced participants' frustration levels relative to a baseline. Although the final adaptive interface used passively recorded data streams, their actual inference required secondary-task user input, thereby negating the passivity of their discovery phase. Whether the discovery phase of the development of an AUI based on operator-state needs to be passively

recorded in order for the resulting inference engine to be considered PDM should be further explored in future research.

Appendix B – CogLog Specifications

The Cognitive Logger (CogLog) was designed to provide a tool for inducing attentional states in users and recording their behaviour while in that state. Streams of this recorded behaviour were then used to develop machine learning classifiers. This purpose is relatively uncommon amongst research programs or platforms and therefore imposed some unique design challenges. The following sections will outline the design requirements of the system for addressing these challenges.

5.7. Design Objectives

5.7.1. Induction

The first task of CogLog is to induce an intended attentional state relative to a baseline (e.g., Attentional tunneling, information overload, etc.). To preserve the validity of this induction process, the stipulation that any induced attentional state is only relative to a baseline is critical. This stems from the fact that there is rarely a clear threshold that defines an attentional state due to the spectrum on which most candidate states are likely to fall (e.g., more or less focused).

Perhaps the most significant design consideration to ensure the validity of an attentional state induction is to ensure that the recorded behaviour that will be classified does not come from different tasks, but is instead derived from different states within the same task. This is critical because the behaviour that we are observing is often task-dependent and will therefore vary significantly across tasks. This problem does not exist to the same extent with physiological measurement because the signal that is being classified is further abstracted from the task that is generating that signal.

A second important experimental design consideration stems from the amount of data that is required to generate a reliable machine learning classifier. For this research program, we had difficulty getting access to expert users. We therefore sought to design tasks that emulated aspects of a more complex task environment, but that would be approachable to a general population, which allowed us to deploy the task on MTurk. The limitations of a user's research program should determine the degree to which this is considered.

5.7.2. Recording

In order to develop a machine learning classifier based on behaviour derived from two attentional states, one must have as much of that behaviour recorded as possible. The current versions of CogLog record:

- User cursor position,
- Canvas position on the screen (i.e., where the center of the canvas falls on the users screen),
- The size of the canvas, and
- The size of the user's browser window.

Based on these three data points we were able to calculate:

- Cursor position on the canvas,
- The exact area of the canvas that is currently in view, and
- The percentage of the canvas that is currently in view.

This data may seem relatively sparse, but it provides the necessary data to fully recreate the experimental trials.

The data recording in Phase 2 was input-dependent. This meant that the logger was triggered every time the user initiated an input. With the input-dependent logs we were then able to convert the dataset to sample at any rate we wished, provided that rate was slower than the maximum rate of the input-dependent method.

In Phase 3 we converted the logging method to sample at fixed rates. We did this because the eventual implemented classifier would have to sample at a fixed rate. It would have been possible to conduct Experiment 3.1 with a input-dependent sampler, determine the optimal rate, and then set up Experiment 3.2 to sample at that rate. This method should be explored in future studies.

There are a variety of other important design considerations that went into the development of CogLog. These considerations will be outlined in greater detail in future publications.

Appendix C – Classifier Development

The level of detail provided in Chapter 3 on the CNN-LSTM used to classify states of attentional tunneling was relatively sparse due to the applied-focus of the venue in which the chapter was published. This appendix will therefore provide additional detail in terms of the actual architecture of the model as well as the process by which it was arrived at.

5.8. Choice of CNN-LSTM

Our initial efforts to classify states of attentional tunneling employed an approach that was essentially identical to what we did for the information overload dataset. We used a combination of TSFresh (Christ, Braun, Neuffer, & Kempa-Liehr, 2018) and random forest classifiers. However, this did not yield successful results, failing to classify the dataset above chance level. This was likely due to the relatively small size of the dataset (~13 hours).

To account for the limited size of the dataset, we explored more complex neural network classification approaches. Our first approach consisted of using basic recurrent neural networks (RNNs). RNNs are well-suited for classifiying time series data (Hüsken & Stagge, 2003), but can suffer from a problem known as the vanishing gradient problem wherein multiple successive components are multiplied by zero, rendering distant temporal learning impossible (Pascanu, Mikolov, & Bengio, 2012). Vanishing gradients prevented our model from being successful, again limiting it to chance performance.

To account for the vanishing gradient problem, we then advanced to Long Short-Term Memory neural networks (LSTM). LSTM networks can represent past information in their internal states and use these states to process new information, but they do not suffer from the vanishing gradient problem that can limit traditional recurrent neural networks (Hochreiter & Schmidhuber, 1997; Lipton, Berkowitz, & Elkan, 2015). LSTMs control how internal states update and output at each time step via gating functions (Hochreiter & Schmidhuber, 1997). This allows them to maintain long term storage of internal states and therefore to exploit distant temporal dependencies within the data.

Stacked LSTM networks are recognized as being state of the art in many time series classification and prediction tasks (Greff, Srivastava, Koutník, Steunebrink, & Schmidhuber, 2017; Prieto, Alonso-González, & Rodríguez, 2015; Zhao, Chen, Wu, Chen, & Liu, 2017). To manage their computational expense, some researchers have stacked multiple LSTM layers under initial 1-Dimensional convolutional layers (CNN-LSTM; Karim, Majumdar, Darabi, & Chen, 2017; Tan et al., 2018). Not only does this improve computation speed, but it also can greatly improve feature extraction capabilities (Pak, Kim, Ryu, Sok, & Pak, 2018).

The use of the CNN-LSTM model yielded performance that was significantly above chance level. This was therefore the final model included in our publication. It should be noted that performance above that observed (0.73 AUC) is likely possible. Future researchers should explore more neural network architectures and data cleaning techniques.