Cognition and Context-Aware Computing: Towards a Situation-Aware System with a Case Study in Aviation

Justin C. Wilson
Southern Methodist University, wilsonj@smu.edu

Follow this and additional works at: https://scholar.smu.edu/engineering_compsci_etds

Part of the Artificial Intelligence and Robotics Commons, and the Graphics and Human Computer Interfaces Commons

Recommended Citation

This Dissertation is brought to you for free and open access by the Computer Science and Engineering at SMU Scholar. It has been accepted for inclusion in Computer Science and Engineering Theses and Dissertations by an authorized administrator of SMU Scholar. For more information, please visit http://digitalrepository.smu.edu.
COGNITION AND CONTEXT-AWARE COMPUTING
TOWARDS A SITUATION-AWARE SYSTEM WITH A CASE STUDY IN AVIATION

A Dissertation Presented to the Graduate Faculty of the

Lyle School of Engineering
Southern Methodist University

in
Partial Fulfillment of the Requirements
for the degree of
Doctor of Philosophy

with a
Major in Computer Science

by
Justin Charles Wilson

B.S., Baylor University, 2006
M.S., Southern Methodist University, 2012
M.S., United States Air Force Test Pilot School, 2015

August 4, 2020
In aviation, flight instructors seek to comprehend the intent and awareness of their students. With this awareness, derived from in-flight observation and post-flight examination, a human instructor can infer the internal contexts of their student aviators as they perform. It is this understanding that is fundamental for evaluating student development. Further, a well-understood construct for describing the state of knowledge about a dynamic environment is known as situational awareness (SA). Often pilot error is associated with SA [80], and it is fundamental to flight safety and mission execution. If these contexts can be automatically inferred, instructors and students can more easily determine when mastery of a particular skill has been achieved — potentially improving the overall quality of student understanding. The goal of this research is the creation of a situation-aware inference system. A system that can evaluate the human-machine state, elucidating cognitive load, situational awareness, engagement, stress, gaze patterns, and the stimulus to attention and performance trade-offs.

In this work, we investigate the automated classification of cognitive load, leveraging the aviation domain as a surrogate for complex task workload induction. We used a mixed virtual and physical flight environment, given a suite of biometric sensors utilizing the HTC Vive Pro Eye and the E4 Empatica. We created and evaluated multiple models. And we have taken advantage of advancements in deep learning such as generative learning, multi-
modal learning, multi-task learning, and X-Vector architectures to classify multiple tasks across 40 subjects inclusive of three subject types — pilots, operators, and novices. Our cognitive load model can automate the evaluation of cognitive load agnostic to subject, subject type, and flight maneuver (task) with an accuracy of over 81%. Further, this approach has been validated with real-flight data from five military test pilots collected over two test and evaluation flights on a C-17 aircraft.

Gaze tracking in the context of learning is a particularly exciting area of research as it can provide a better understanding of intent, situational awareness, and attention. We investigate how gaze tracking in a mixed virtual and physical flight environment can be employed to mimic the observations of a flight instructor. Our gaze classifier focuses on the classification of gaze scan patterns for aviators in a mixed-reality flight simulation. We created and evaluated two models: a task-agnostic model and a multi-task model. Both models use deep convolutional neural networks to classify the quality of pilot gaze scan patterns as compared to visual inspection by three experienced flight instructors. Our multi-task model can automate the process of gaze inspection for three separate flight tasks.

Our approach could assist existing flight instructors throughout the learning process, inform operational success and failures, or it may even open the door to more automated feedback for pilots by improving the execution of various phases of flight.
# TABLE OF CONTENTS

LIST OF FIGURES ................................................................. xi
LIST OF TABLES ................................................................. xiv

CHAPTER

1. INTRODUCTION ............................................................... 1
   1.1. Contributions ........................................................... 2
   1.2. Structure of Dissertation ........................................... 5

2. CONTEXT-AWARE COMPUTING .......................................... 6
   2.1. What is Context? ...................................................... 7
   2.2. Context-aware Computing ........................................... 12
   2.3. Categorical Context .................................................. 17
       2.3.1. Conceptual and Operational ................................. 18
       2.3.2. Latent Context .................................................. 19
   2.4. Conclusions ............................................................ 22

3. DEEP LEARNING .............................................................. 24
   3.1. Generative and Discriminative Models ......................... 24
   3.2. Transfer Learning .................................................... 25
   3.3. Multi-task Learning ................................................... 26
   3.4. Multi-modal Learning ............................................... 26
   3.5. Variational Autoencoder ............................................ 28
   3.6. X-Vector Inspiration ............................................... 29
   3.7. Conclusions ............................................................ 30

4. SITUATION AWARENESS, COGNITIVE LOAD, AND LEARNING ......... 31
   4.1. Situation Awareness .................................................. 32
8.3.4.3. Fixed-Length BM^3TX Training ................................. 105

8.3.5. Variable-Length Input BM^3TX ................................. 106
8.3.5.1. Variable-Length BM^3TX Training ............................. 108
8.3.5.2. Model Training Discussion .................................... 108

8.4. Results .............................................................. 109
8.4.1. Initial Models - Multiple Ratings Domain ....................... 110
8.4.2. Multiple Domain vs Single Domain Models ....................... 113
8.4.3. Multi-Task vs Single Task Models ............................... 114
8.4.4. Superior Domain Family .......................................... 115
8.4.5. Modality Combination Performance .............................. 117
8.4.6. Performance Given Subject Experience .......................... 117
8.4.7. Discussion ......................................................... 120

8.5. Edwards Pilot Workload Study ................................. 121
8.5.1. Results .............................................................. 121

8.6. Conclusions .......................................................... 122

9. THE CLASSIFICATION OF GAZE SCAN PATTERNS .............. 124
9.1. Introduction and Motivation ....................................... 124

9.2. Background and Related Work .................................... 127
9.2.1. Sight Picture ......................................................... 127
9.2.2. Heatmaps in Eye-Tracking ....................................... 127
9.2.3. Gaze in Aviation .................................................. 129
9.2.4. Other Works in Gaze Classification ............................ 130
9.2.5. Expert Review ..................................................... 131

9.3. Gaze Quality Labeling .............................................. 132
9.3.1. Review Interface Creation ....................................... 133
9.3.2. Rating Process ..................................................... 134
9.3.3. Inter-Rater Reliability Results ......................................................... 134
9.3.4. Multi-task Labeling ................................................................. 136
9.4. Deep Learning Architecture ......................................................... 137
  9.4.1. Input Heatmap Generation ......................................................... 139
  9.4.2. Training ................................................................. 140
9.5. Data Augmentation ................................................................. 140
9.6. Results ................................................................................. 142
  9.6.1. Task Agnostic Model ................................................................. 142
  9.6.2. Multi-task Model ................................................................. 143
  9.6.3. Human-Model Inter-Rater Reliability ......................................... 145
9.7. Heatmap Examples .................................................................. 146
9.8. Model Training Hardware and Software ...................................... 147
9.9. Discussion .............................................................................. 148
9.10. Conclusions .......................................................................... 151
10. CONCLUSIONS ........................................................................ 153
  10.1. Contributions ........................................................................ 154
  10.2. A Way Forward .................................................................... 155
APPENDIX
A. BEDFORD WORKLOAD SCALE ............................................................ 158
B. NASA TASK LOAD INDEX RATING .................................................. 159
C. SAGAT QUESTIONNAIRE ............................................................... 161
D. BIOMETRIC CAPTURE SOFTWARE .................................................. 163
  D.1. JSON Format ........................................................................ 163
  D.2. Screen Shots ....................................................................... 166
BIBLIOGRAPHY ........................................................................... 167
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Perera et al.’s [188] context categorization in two different perspectives: conceptual and operational.</td>
</tr>
<tr>
<td>3.1</td>
<td>Transfer Learning Example</td>
</tr>
<tr>
<td>4.1</td>
<td>Attributes of cognitive load [179]</td>
</tr>
<tr>
<td>4.2</td>
<td>Hypothetical tracking performance given decreasing allowable error [100]</td>
</tr>
<tr>
<td>5.1</td>
<td>An example of the components of an ER-SCR [56]</td>
</tr>
<tr>
<td>6.1</td>
<td>Boom Limits Demonstration — Points five and six are inner and outer limits [127]</td>
</tr>
<tr>
<td>6.2</td>
<td>Components of the pulse wave [76]</td>
</tr>
<tr>
<td>6.3</td>
<td>L3Harris Feasibility Study PPG, IBI, and Acceleration: Magnitude baseline, participant six</td>
</tr>
<tr>
<td>6.4</td>
<td>L3Harris Feasibility Study EDA: Final approach and landing, participant two</td>
</tr>
<tr>
<td>6.5</td>
<td>L3Harris Feasibility Study EDA: Final approach and landing, participant three</td>
</tr>
<tr>
<td>6.6</td>
<td>L3Harris Feasibility Study ER-SCRs: Oddball baseline scenario correlated to oddball tones, participant three</td>
</tr>
<tr>
<td>6.7</td>
<td>Edwards Objective Measures Study Examples of acceleration magnitude: (Top) Pilot A, Flight 1, AR boom limits (Bottom) Pilot D, Flight 2, AR boom limits</td>
</tr>
<tr>
<td>6.8</td>
<td>Edwards Objective Measures Study PPG, IBI, and Magnitude</td>
</tr>
<tr>
<td>6.9</td>
<td>EDA tonic — AR boom limits demo</td>
</tr>
<tr>
<td>6.10</td>
<td>Edwards Objective Measures Study EDA, SCL, and SCR</td>
</tr>
<tr>
<td>6.11</td>
<td>Pilot A: EDA on Takeoff</td>
</tr>
<tr>
<td>7.1</td>
<td>BBXR mixed-reality simulator [110]</td>
</tr>
<tr>
<td>7.2</td>
<td>Air Intercept Diagram</td>
</tr>
</tbody>
</table>
8.1 A comparison of corrected rating domains and their respective inter-subject alignment (10 quantiles): (a) Equal Weight NASA-TLX, (b) NASA-TLX Mental Demand sub-scale, and (c) Bedford Workload 

8.2 Label distributions for classifier tasks

8.3 Example PPG variational autoencoder architecture

8.4 Variable-Length $BM^3TX$: BlandAltman plot for regression, including (a) all eight, (b) test set, and (c) one fold

8.5 Power Set: Average Single- vs Multi-Task Accuracies

8.6 Binary accuracy — family of models for all three domains of the variable-length $BM^3TX$ architecture: (a) RTLX, (b) Mental Demand, (c) Bedford Workload

8.7 Equal weight RTLX family of models, ordered by average fold accuracy, all subjects

8.8 Equal weight RTLX family of models, ordered by average fold accuracy and trained only on: (a) Novices, (b) Operators, (c) Pilots, and (d) All subjects

9.1 Aggregate high-resolution heatmap of a normal landing

9.2 Example of a concatenated video frame from a video that instructors reviewed for labeling gaze quality. Three heatmaps of different screenspaces line the top of the frame and the bottom consists of video from the pilot (left) and a zoomed heatmap of the HUD (right).

9.3 Sample counts for task agnostic and multi-task models

9.4 Four example center screenspaces (a-d) — While (d) would rate “poor” for all 3-tasks, (a-c) are each characterized as (a) “correct”, (b) “fair”, and (c) “poor” given the climb, cruise, and decent task. However, (b) is labeled “correct” for the ground task, as only airspeed and centerline are needed, and (c) is labeled “correct” for the final approach task, a pilot flying the AoA bracket.

9.5 Task Agnostic Convolutional Neural Network

9.6 Multi-task Convolutional Neural Network

9.7 Center Screenspace Scaled 25 X 25 (A) Count Matrix, (B) Heatmap, (C) Gaussian Blur, and (D) Actual 244 x 244

9.8 Agnostic task model: (a) Combined confusion matrix over 10-folds (b) Confusion matrix over test dataset
9.9 Multi-task model combined confusion matrices over: (a) 10-folds climb, cruise, and decent, (b) 10-folds ground, (c) 10-folds final approach (d) Climb, cruise, and decent test dataset, (e) Ground test dataset, and (f) Final approach test dataset .......................................................... 144

9.10 CCD task examples: poor (left), fair (right), and correct (bottom) .................. 147

9.11 Ground task: poor (left), fair (right), and correct (bottom) ......................... 147

9.12 Final approach task: poor (left), fair (right), and correct (bottom) .............. 148

9.13 Negative heatmap examples: all heatmaps are rated 'poor' for CCD, ground, and final approach. All cover a 30-second window period ..................... 149

9.14 Aggregate example heatmaps for final approach: (a) zoomed in heatmap of HUD, (b) Aggregate real-life flight on final approach in a C-17 aircraft [154] 151

A.1 Bedford Workload Scale ............................................................... 158

B.1 NASA Task Load Index Rating Sheet Unweighted (RTLX) ...................... 159

B.2 NASA Task Load Index Rating Definitions ..................................... 160

C.1 Alpha Questionnaire ................................................................. 161

C.2 Bravo Questionnaire ............................................................... 162

D.1 Capture Software Screen Shots .................................................. 166
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>L3Harris Feasibility Study Maneuvers</td>
<td>60</td>
</tr>
<tr>
<td>6.2</td>
<td>EDA Filter Coefficients</td>
<td>65</td>
</tr>
<tr>
<td>6.3</td>
<td>L3Harris Feasibility Study Magnitude Statistics</td>
<td>67</td>
</tr>
<tr>
<td>6.4</td>
<td>L3Harris Feasibility Study IBI Statistics: Baseline secondary task, participant six</td>
<td>68</td>
</tr>
<tr>
<td>6.5</td>
<td>L3Harris Feasibility Study EDA Statistics: Final approach and landing, participant two</td>
<td>69</td>
</tr>
<tr>
<td>6.6</td>
<td>L3Harris Feasibility Study EDA statistics: Final approach and landing, participant three</td>
<td>70</td>
</tr>
<tr>
<td>6.7</td>
<td>L3Harris Feasibility Study Peak SCRs — All Subjects</td>
<td>71</td>
</tr>
<tr>
<td>6.8</td>
<td>Edwards Objective Measures Study Magnitude Statistics</td>
<td>73</td>
</tr>
<tr>
<td>6.9</td>
<td>Edwards Objective Measures Study IBI statistics for climb, Pilot E</td>
<td>75</td>
</tr>
<tr>
<td>6.10</td>
<td>Edwards Objective Measures Study Peak SCRs — All Subjects</td>
<td>78</td>
</tr>
<tr>
<td>7.1</td>
<td>Subject Demographics</td>
<td>85</td>
</tr>
<tr>
<td>8.1</td>
<td>VAE Architecture: The VAE configurations for each modality. A layer’s parameters are denoted as ‘Layer Type $&lt;\text{KernelSize}&gt; -$ $&lt;\text{Strides}&gt;$ $- &lt;\text{Filters}&gt;$ $- [\text{Activation Type (ReLu if not noted})]$.’</td>
<td>102</td>
</tr>
<tr>
<td>8.2</td>
<td>VAE training test dataset results</td>
<td>103</td>
</tr>
<tr>
<td>8.3</td>
<td>Modality Alignment Parameters</td>
<td>104</td>
</tr>
<tr>
<td>8.4</td>
<td>$BM^3TX$ Architectures: A layer’s parameters are denoted as ‘$&lt;\text{Dense Nodes}&gt;$ $- [\text{Activation Type (ReLu if not noted})]$.’</td>
<td>105</td>
</tr>
<tr>
<td>8.5</td>
<td>Architecture fold metrics for each subjective ratings domain — single model</td>
<td>110</td>
</tr>
<tr>
<td>8.6</td>
<td>Architecture fold metrics for variable-length $BM^3TX$ — a separate model for each subjective ratings domain</td>
<td>114</td>
</tr>
</tbody>
</table>
8.7 Variable-length $BM^3TX$: Metrics from the classification of Edwards Pilot Workload Study flight data — five test pilots across two flights flown August 28th and August 30th, 2018. .......................... 122

9.1 Rater Experience ................................................................. 131

9.2 Inter-rater Reliability ............................................................ 136

9.3 Inter-rater Reliability, Model as Additional Rater ...................... 145
I dedicate this dissertation to my mother and father for their unconditional love and support. And to my wife, who has steadfastly supported me through this endeavor.
Context-aware computing is an operative force that facilitates ubiquitous computing. A force Mark Weiser [275] sought in “a computer for the 21st century.” A computer that is adept in understanding context through pervasive and ubiquitous sensors. And through context, a machine capable of response, projection, and adjustment in support of the human-in-the-loop. Further, with the advancement of computer and adjacent technologies, a plethora of sensing modalities are emerging, getting smaller, and more mobile. As these devices have continued to proliferate and become more omnipresent, a substantial opportunity for furthering the understanding of the context of the human-in-the-loop has arrived.

In the context of aviation, a well-understood construct for describing the state of knowledge about a dynamic environment is the situation(al) awareness (SA). Ensuring accurate SA is fundamental to basic flight, mission success, and safety. The lack or loss of SA is frequently connected with pilot error [81,90] because SA is linked probabilistically with pilot performance [79].

When training, the performance of aviators is currently assessed through instructor observation and review. Beyond flying ability and aircraft response, observation can include visual signs of emotion and stress, aggressiveness, speech, pilot gaze and scan patterns, and response delay. Through these observations, instructors can infer the internal context of aviators while they learn, which is essential for assessing their development of multiple skills. While understanding this internal context is recognized as important, the automated sensing of this context is not yet possible. Context- or situation-aware avionics — capable of assisting either the flight instructor or the aviator in understanding the human-machine state [36] in real-time — is highly sought after. By inferring learner context automatically, instructors and trainees can more easily determine when mastery of a particular skill is
achieved, expediting the learning process, and potentially improving the overall quality of learner understanding.

The activities of this research involved the collection of physiological measures relevant to aviators, such as pilots and operators, while they undergo a simulated flight scenario. With this information we seek to understand the latent context behind the internal states of the pilot using deep multi-modal and multi-task learning. Ultimately, the main goal is to create an inference system that can perform assessments by reliably measuring situational awareness and cognitive load of a pilot in near real-time using physiological sensors.

The opportunities for measuring human performance via biometric sensors is hard to overstate. Four crucial ideas have emerged: (1) The potential to accelerate learning through additional insights regarding when an aviator has or has not mastered a particular exercise, including what elements are most confusing or cognitively complex. (2) The potential to relate operational successes of aviators to their collected human performance data. In this way, the system may allow the inference of an aviators susceptibility to different types of error. (3) The proposed system could inform the design of real-time assessment systems for aviators outside of simulation environments, which is of interest to a number of stakeholders throughout aviation. And finally, (4) adaptive user interfaces adjusting themselves based upon the inferred internal state of the user — situational awareness, attention, cognitive load, emotion, stress, engagement, etc.

1.1. Contributions

This research introduces the concept of “latent context, Section 2.3.2, as a third operational category of context. Latent context is learned by modeling the latent underlying distribution of contextual information, rather than having been looked up or derived, analytically, or through the combination of other sources. Additionally, latent context can be learned through the use of the many advances in machine learning, including multi-task and multi-modal deep learning, and generative models. With an understanding of context and context-aware computing, three human subjects experiments were conducted:
1. A feasibility study with ten subjects conducted on an aircraft simulator in partnership with L3Harris.

2. A five-subject (test pilots) study conducted in real flight at Edwards AFB supported by the USAF.

3. A full 40-subject data collection effort conducted on a mixed-reality simulator in partnership with L3Harris.

Throughout all three experiments, subjective workload ratings were collected with NASA Task Load Index (NASA-TLX) [113] and Bedford Pilot Workload Scale [208]. Using this data, several machine learning models were built. Further, expert instructors in aviation were recruited to evaluate and rate pilot scan patterns. In this research, we propose methods and architectures for the classification of cognitive load and pilot gaze scan patterns.

The contributions presented in this dissertation are listed as follows:

- We introduced the concept of latent context as a third operative category of context (Section 2.3.2)

- We designed and carried out multiple human subjects experiments for aviators in a variety of flight scenarios, as reviewed in Chapters 6 and 7. Biometric data, Chapter 5, were collected for five modalities across six low and high workload maneuvers.

- We proposed a novel subjective label alignment scheme using subjective ratings for cross-subject cognitive load classification, discussed in Section 8.2.1

- We investigated three competing, machine learning architectures for the classification of cognitive load: (1) a multi-modal, feature engineered random forest classifier, (2) a fixed-length input deep learning classifier, and (3) a variable-length input deep learning classifier. The deep learning models take advantage of generative learning, multi-modal learning, multi-task learning, and X-Vector like architectures. Our binary, multi-modal model can automate the evaluation of cognitive load with an accuracy of over 81.0% agnostic to the subject, and to flight maneuver.
• We validated our cognitive load deep learning approach with real-world data from five test pilots collected over two military test and evaluation flights on a C-17 aircraft (CL > 90.0% accurate with a few caveats).

• We proposed methods for novel gaze data augmentation specific to pilot scan patterns that increase the robustness of trained machine learning models (Section 9.5).

• We investigated two competing, convolutional neural network architectures: a task agnostic model and a multi-task model (Section 9.4) for the classification of pilot gaze patterns. We evaluate the architectures with K-fold cross-validation, achieving greater than 93.0% average test accuracy compared to instructor observation (Section 9.6).

• We further analyzed gaze scan pattern inter-rater reliability (IRR) amongst raters and classifiers finding strong agreement; and in some cases, machine-to-human IRR is higher than human-to-human IRR, Section 9.3.
1.2. Structure of Dissertation

The remaining sections of this proposal contain the following content:

In Chapter 2, we present the background of context and context-aware computing and latent context is introduced. Chapter 3 elucidates the important background for learning latent spaces and building machine learning models. Chapter 4, discusses the related background in various sub-disciplines related to SA, gaze, cognitive load, and applications in aviation. Chapter 5 discusses the sensing modalities used throughout this research. In Chapter 6, the two feasibility studies, simulation and real-flight, are described and the results of both experiments are discussed. Chapter 7 presents the full 40-subject study, the data collection effort, used in the development of our machine learning models. Chapter 8 explores the development and investigation of the three competing machine learning architectures for the classification of cognitive load. Chapter 9 explains the development of the gaze pattern classifiers and the results, and portions of this chapter are published in: J.C. Wilson, S. Nair, S. Scielzo, and E.C. Larson (August 2020). Automatic Gaze Classification for Aviators: Using Multi-task Convolutional Networks as a Proxy for Flight Instructor Observation. International Journal of Aeronautics, Aviation, and Aerospace (IJAAA). Finally Chapter 10 provides a review of this research and a way towards continued development of a situation-aware system.
Chapter 2
CONTEXT-AWARE COMPUTING

Computer technology is pervasive in society, due equally to the explosive growth in connectivity from the internet and the miniaturization of computing hardware. This widespread use of computing is often referred to as Ubiquitous Computing, where many interconnected devices form a larger system. These devices have many sensing capabilities so that each device can add unique information about the environment and person using it when employed in the larger system. As these devices continue to proliferate and become more omnipresent, more sensing modalities are developed and become available — leading to significant potential for understanding the context of the human-in-the-loop, or even the pervasive entities-in-the-loop. These developments were implied by Mark Weiser in his landmark paper on ubiquitous computing [275]. However, what enables this technology is truly intriguing.

While Mark Weiser, father of Ubicomp [275], did not define context-aware computing, as that would come later, [220], he did allude to it. He sought “a computer for the 21st century” capable of inferring context through pervasive sensors and proficient in responding, projecting, and adjusting to the dynamics of that context. While becoming invisible and un-intrusive to the “common awareness” — “specialized elements of hardware and software, connected by [any medium] will be so ubiquitous that no [human] will notice their presence” [275]. Context-aware computing technology can be viewed as an operative force that enables ubiquitous computing [223, 243]. But what is “context?” And what is “context-aware computing?” These are questions the research community has been trying to answer since Schilit and Theimer coined the term context-aware computing [221].
2.1. What is Context?

Over the years a general definition of “context” has been difficult to fully clarify; computing systems have become more complex and their sensing capabilities far more reaching. Early on, context was particular and defined regarding the singular application and task at hand. These application-specific definitions were thought to be limiting, and a more generalized definition was sought. Even today, there is a prevalent feeling that current definitions do not fully generalize in scope to encompass every aspect of context. Researchers continue to improve upon such a definition. In this chapter, we discuss that evolution. There were many early definitions of context. Context was described as locations, identities of nearby people and objects, and changes to those objects [221]. Schilit, Adams, and Want explained it in terms of where you are, who you are with, and what resources are nearby [220]. Dey, Abowd, and Wood defined context as any information about the user and the environment that can be used to enhance the user’s experience [61]. Schmidt, Beigl, and Gellersen noted, context is a key issue in the interactions between humans and computers, describing the surrounding facts that add meaning [223]. Pascoe described context as a subjective concept that is defined by the entity that perceives it [29]. Note that Pascoe’s definition does not necessarily describe context with respect to the user. Dey and Abowd took issue with early definitions citing several flaws [61]. They described some as context by example, [29, 212, 221]. Others were noted as synonyms for context, [27, 86, 119, 203, 272]. And they found several definitions to be too specific, [62, 186, 220], arguing that context must consider the whole situation relevant to an application and its set of users. Enumerating every aspect of all situations is simply not possible [61]. They argued that information not relevant to the situation with respect to the interactions and the assemblage of users is not context. If a set of data or knowledge characterizes the situation of a participant within a given interaction, then that information is context [60]. Dey and Abowd sought a more operative definition that generalizes well; this culminated into the most broadly used definition of context in the literature: Context is re complex and
their sensing capabilities far more reaching. Early on, “context” was particular and defined regarding the singular application and task at hand. These application-specific definitions were thought to be limiting, and a more generalized definition was sought. Even today, there is a prevalent feeling that current definitions do not fully generalize in scope to encompass every aspect of context. Researchers continue to improve upon such a definition. In this chapter, we discuss that evolution.

There were many early definitions of context. Context was described as “locations, identities of nearby people and objects, and changes to those objects” [221]. Schilit, Adams, and Want explained it in terms of “where you are, who you are with, and what resources are nearby” [220]. Dey, Abowd, and Wood defined context as “any information about the user and the environment that can be used to enhance the users experience” [61]. Schmidt, Beigl, and Gellersen noted, “context is a key issue in the interactions between humans and computers, describing the surrounding facts that add meaning” [223]. Pascoe described context as “a subjective concept that is defined by the entity that perceives it” [29]. Note that Pascoes definition does not necessarily describe context with respect to the user.

Dey and Abowd took issue with early definitions citing several flaws [61]. They described some as “context by example”, [29, 212, 221]. Others were noted as synonyms for context, [27, 86, 119, 203, 272]. And they found several definitions to be too specific, [62, 186, 220], arguing that context must consider “the whole situation relevant to an application and its set of users.” Enumerating every aspect of all situations is simply not possible [61]. They argued that information not relevant to the situation with respect to the interactions and the assemblage of users is not context. If a set of data or knowledge characterizes the situation of a participant within a given interaction, “then that information is context” [60].
Dey and Abowd sought a more operative definition that generalizes well; this culminated into the most broadly used definition of context in the literature:

**Context** is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [61].

Seeking a definition from a wholly-mobile computing perspective, Chen and Kotz defined context by delineating between the concepts of determining behavior and relevance to the user. Specifically, they described context as “the set of environmental states and settings that either determines an applications behavior or in which an application event occurs and is interesting to the user” [42]. Further, they categorized context as active or passive. Active context is information “that influences the behaviors of an application,” and passive context is information “that is relevant to, but not critical to an application” [42].

There have been other perspectives and attempts to define context. Most notably, Paul Dourish, who desired both a clarification of and a foundation for context-aware computing [69, 70], stated that there are two points of view on what context is, a positive or representational view and a phenomenological or interactional view. Specifically, a representational view of context makes four assumptions: 1) context is a form of information and it can be encoded; 2) context is separable and can be defined in advance; 3) context is stable; that is context does not change among instantiations; and 4) context and activity are separable. Dourish argues these assumptions are inaccurate because the positive view approaches accounts of social behavior that are separate from the observer. With agency and interpretation as central roles to all social behavior for phenomenological theories, positive accounts of social action cannot be separate or independent of the observer. Accounts of social behavior are in themselves outcomes of social behavior. Therefore, Dourish offered four assumptions for the interactional view, which follow. 1) Contextuality is a relational attribute among objects and/or activities; it is relevant with respect to the activity/object. 2) The scope of context is defined dynamically as opposed to being defined in advance. 3) Context is not stable, rather it is an occasional attribute relevant to the particular in-
stance of action and the entities involved. 4) Context and content are not separable; context emerges from the activity. Context is “not a representational problem but [an] interactional problem” [70]. His definition of context is as follows:

**Context** is an occasioned property of action. It is managed moment-to-moment, achieved by those carrying out some activity together, and relative to that activity and to the forms of action and engagement that it entails [70].

Or more simply, **Context** is a set of descriptive features of setting in the paradigm of practice [70]. Where practice is a process in which one can experience our world, and through their engagement with it, then the experience becomes meaningful [273].

Henricksen argued for a clear distinction among context, context models, and context information. Pointing out that the idea of context is nebulous at best, making it troublesome to define. She asserted that context should be considered relative to the task and not between “the interactions of the users and applications” [116]. Context is the apparatus for applications to accomplish tasks autonomously in a flexible manner for the user. She defines context, context models, and context information as follows:

*The Context of a task is the set of circumstances surrounding it that are potentially of relevance to its completion* [116].

A **context model** identifies a concrete subset of the context that is realistically attainable from sensors, applications and users, and able to be exploited in the execution of the task. The context model that is employed by a given context-aware application is usually explicitly specified by the application developer but may evolve over time [116].

**Context information** is a set of data, gathered from sensors and users, that conforms to a context model. This provides a snapshot that approximates the state, at a given point in time, of the subset of the context encompassed by the model [116].
More recent surveys such as from Baldauf, Dustdar, and Rosenberg [16] identified how challenging the task is to define context, arguing in favor of Dey and Abowd's definition. Perera et al. and Alegre et al. [6, 188], acknowledged Dey and Abowd's definition is the most widely accepted. Perera et al. noted that “context implicitly provide[s] the meaning of information” with respect to Dey and Abowd's definition. However, they espoused the importance of identifying “context from data in general,” making a distinction between the data element and context, referencing Sanchez et al.'s work. Context information is produced by processing raw sensor data, verifying consistency, and adding metadata [217]. This is a somewhat similar concept to Henricksens without discussing the context model. While, Dey and Abowd's definition of context generalizes well and is the most widely accepted, clearly context can be defined from more than one vantage point.

In our research, while we tend to agree mostly with Dey and Abowd's definition. We prefer a modified version given the thoughts of the multiple researchers discussed above and our own:

**Context** is any information that can be used to characterize the situation of an entity.

An entity is a person, place, or object that is considered relevant to the interaction among entities, including the entities themselves.

Here 'entities' have replaced 'user and application.' We believe context is information surrounding the interactions between entities and themselves. We agree that information is only context when it is relevant to the situation concerning the interactions and the assemblage of users. We ask, why just human users? Can an entity be a user? Pascoe defined context with respect to an entity, as did Dey and Abowd. However, Dey and Abowd caveated its dependence on an interaction between a user and an application. We found this limiting. Further, given Dey and Abowd's definition, a user and an application are both entities. We argue that it is useful to understand the context of interactions that are not necessarily a human and an application. With the present level of technology, two adaptive applications that are black boxes with respect to each other can potentially make use of the context that exists between them. This would likely be in support of a more intricate task that enables a
more complex goal of a human or even multiple human entities. Further, it is very interesting to understand the context that exists among interacting human entities or even human entities with other entities — especially in the social domain. This new definition enables the understanding of context given the interactions between heterogeneous entities. Therefore, we consider a set of information to be context if this information characterizes the situation of an entity within a given interaction. Conversely, if information or a set of information is not relevant, and it does not characterize an interaction, then it is not context. This discussion will continue in the following section.

2.2. Context-aware Computing

One of the key reasons the literature does not agree on context is that there is not complete agreement on the true nature of context-aware computing. As with context, there are many definitions of context-aware computing; here we discuss several focusing on its evolution.

First, Schilit, Theimer, and Welch called context-aware computing the “dynamic customization of applications” [222]. They noted its capability to change the application at any time, such as in response to varying shifts in the “physical computing environment.” Spreitzer and Theimers paper [243] notes that, if the information is made available to an individual’s computing environment; where this information provides the who and the what that is “in the vicinity of a person.” Then applications can be made sensitive to that person’s context and respond accordingly. Following this thread, Schilit and Theimer ultimately called this capability context-aware computing [221].

Schilit, Adams, and Want [220] described context-aware computing as “a class of applications that are aware of the context in which they run.” They defined context-aware software as “[adapting] according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time.” They characterized context-aware computing as systems capable of examining their surroundings and reacting to changes to those surroundings.
Dey, Abowd, and Wood [62] desired a ubiquitous computing infrastructure that seeks out the user “when and where it is wanted.” They called this context-aware integration because it takes advantage of the potential for integration via shared information dynamically based on user actions.” The user would thus be presented access to services and data, regardless of source – desktop, mobile, or web, based on their relevance to the users current context.

While Schmidt, Beigl, and Gellersen [223] didn’t specifically define context-aware computing, they argued that mobile computing had the potential to benefit from context-aware computing. Specifically, this included the ability to adapt to changes in the environment and to improve the user interface given the context. They described the interaction with applications on mobile devices as being spontaneous and brief. Applications are used intermittently in various locations without closing the application; this prompted the need for context-awareness to provide the mobile device the ability to obtain information about the physical environment and supply it to adaptive applications. Further, they posited that context-awareness can assist an application in controlling the flow of information to the user and in enhancing the content provided to the user, thus improving human-computer interaction (HCI). They provided the example of an adaptive, context-aware interface that modified its orientation based upon the physical orientation of the mobile device – a now common implementation amongst modern smartphones.

Hull, Neaves, and Bedford-Roberts [119] described situated computing as “the ability of computing devices to detect, interpret, and respond to aspects of the users local environment.” P. J. Brown [28] defined context-aware applications as “applications that automatically provide information and/or take actions according to the users present context, as detected by sensors.” Pascoe [186] defined context-awareness as “the ability of a program or device to sense various states of its environment and itself.”

Pascoe described “context-awareness independently of the application, function, or interface” by breaking context-awareness into four categories: 1) context sensing, 2) contextual adaptation, 3) contextual resource discovery, and 4) context augmentation. Where context sensing is the machine’s ability to discern “various environmental states and present[ing]
them to the user.” Context adaptation is described as the ability of applications to leverage their “behavior to integrate more seamlessly with the user’s environment” – effectively tailoring itself to the situation at hand. Contextual resource discovery implies the use of information about the computer’s own context to discover other resources within the same context and to use them to the user’s advantage. Lastly, contextual augmentation is “the augmentation of the environment [reality] with additional information.” For example, updating the user on surrounding attractions based on their current location [186].

Dey and Abowd observed an overarching theme with context-aware computing up to this point. Various definitions fit into two categories: “uses context” [59, 62, 119, 186, 212, 214] and “adapts to context” [28, 29, 220, 272]. Specifically, “uses context” implies that computing devices can “detect and sense” contextual information in order to use, respond to, and provide services to the user. “Adapts to context” implies that computing devices can change itself or its operations/behavior in some way due to changes in context [61]. Endeavoring to balance the two categories of definitions, an operative definition was sought that would be inclusive of existing context-aware applications and would generalize well for future systems. What was needed was a definition that focused on the user and covered even the most simplistic of context-aware systems, i.e., one that simply presented context to the user. They found that by only requiring “response to context”, which fits in both categories, they could sufficiently generalize the definition. Requiring an application to modify its behavior, sense, and/or detect was simply too restrictive for a broad and general definition. Dey and Abowd defined context-aware computing as follows:

A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task [61].

In Chen and Kotz’s context-aware mobile computing survey [42] they discussed the thread of research describing context-aware computing features from Schilit [220], Pascoe [186], and Dey and Abowd [61]. Ultimately, they combined the ideas and argued that there are only two ways in which to use context: 1) automatically adapt application behavior with respect to context, and 2) present the discovered context to the user dynamically or save
the information for later use; this is very similar to Dey and Abowd’s ‘adapts to context’ and ‘uses context’ [61]. With this in mind and the concepts of active and passive context discussed earlier, Chen and Kotz defined context-aware computing for mobile applications with two definitions [42]:

**Active context awareness**: an application automatically adapts to discovered context, by changing the application’s behavior.

**Passive context awareness**: an application presents the new or updated context to an interested user or makes the context persistent for the user to retrieve later.

Dourish [69] sought to extend context-aware computing to include embodied interaction. He explained that ubiquitous computing and tangible computing’s use of context are essentially two sides of the same coin. Stating that phenomenology should be used as a basis for the design and evaluation of context-aware technologies, Dourish suggested there is a broader purpose for context in interaction. Specifically, “the context in which actions take place is what allows people to find it meaningful.” Therefore, the actions themselves are context, as both are “mutually constitutive” [70]. Further, social, organizational, and cultural context along with physical context are key pieces of information in shaping action, given they provide users the information necessary to interpret and understand action. “The embodied perspective claims that actions and their settings are fundamentally intertwined – without the context, there is not action.” Thus, Dourish’s embodied interaction is concerned with the availability for engagement [70]. From this perspective, it is the meaningfulness of ‘artifacts’ that arise out of use within systems of practice, 2.1. A key point is that the concept of practice joins action and meaning; and through practice, context and action evolve, such that the relevance of context and action change, fading in and out, as required. Over time this can bring into focus important fringe details about a task, such as when a novice leather worker evolves into a master craftsman [70]. The way in which he or she looks at and engages with the practice of leather working evolves as the individual develops into a master. Much in the same way a novice research assistant develops into a distinguished
professor. Dourish contended that context-aware systems must support not only a given state of context, but also the evolution of that interaction in practice. Users must be able to “negotiate and evolve systems of practice and meaning” over the period of interaction. Dourish argued that users, not designers, define meaning to their interactions with systems as their “working practices evolve,” then designers must develop systems for which 1) they are able to present aspects of their own context for interpretation by the user, 2) context and content must be fluid – continually available for inspection and change, and 3) human activity must drive the emergence of the structure “that makes information meaningful” [70].

Henrickson described context-awareness as “the exploitation of context information by applications in order to reduce reliance on user input and promote adaptation to dynamic factors in computing environments and user requirements” [116]. Baldauf explained that context-aware or sentient systems “are able to adapt their operations to the current context without explicit user intervention and thus aim at increasing usability and effectiveness by taking environmental context into account” [16]. Pera et al. and Alegre et al. noted the acceptance use of Dey and Abowd definition of context-aware systems. Pera et al. generally agrees with Dey and Abowd in observing its ease of use in disambiguation. They stated that it is important for “context awareness frameworks [to] support acquisition, representation, delivery, and reaction” [188]. Alegre et al. noted the difficulty in defining context-awareness given its dependence on defining context itself [6]. Finally, Temdee and Prasad described context-awareness of a system as the representation of the ability of a system to use context and provide the proper response to users [255].

As with the definition of context, we tend to agree mostly with Dey and Abowd’s definition [61]. Our preference leans towards generalization and usability. An entity that does something with context in support of a user would be an apt, less formal definition, possessing a clear understanding of what context-aware computing is. While we do not need to change the original definition outlined by Dey and Abowd, we again ask, can a user be a non-human entity? We believe it can; one that in support of a task, enables a more complex goal or task of a human or multiple humans. This new interpretation supports the general-
ization of context-aware capabilities throughout the many heterogeneous entities that may exist in a complex system. Therefore, we consider an entity to be context-aware if that entity uses context to provide services and/or information to human and nonhuman entities alike given their relevance towards that goal. The understanding of a complex system, as a whole inclusive of humans and nonhuman entities, is an important research topic especially in the domain of aviation. While the focus of this research is to understand the internal state of a human in a complex system, we see the human as a complex entity within the context-aware and ubiquitous paradigm.

2.3. Categorical Context

This section discusses categorical context, a classification of context that helps designers and developers determine the most useful context for their application [61]. First, it is important to highlight a distinction between raw data and context; Sanchez et al. provides an appropriate distinction [217]. Raw data is unprocessed, “taken directly from the source,” and context is processed raw data. Context is “checked for consistency” and meta data is added as necessary. Further, it includes any derived context; and this refers to context that is calculated from other context and/or raw data, i.e., a distance calculated from two geographic points — provided that distance is relevant to the interaction between a user and the system, but the origin of context is as important as accuracy.

Over the years, numerous researchers have identified types or categorizations of context directed by their purposes or points of view. Perera et al. [188] discussed the various methods at length. Briefly, Schilit et al. [220] categorized context based on three questions: “where you are? who you are with? and what resources are nearby?” Dey and Abowd [61] defined context in terms of location, identity, time, and activity, and designated these conceptual categories as primary context. They further defined a secondary context as all other types of context not listed as primary or rather information that can be discerned given primary context. For example, with a person’s identity (primary) their email address (secondary) can be looked up, such as in key-value pair. Perera et al. pointed out that this categorization
limits one’s ability to clearly understand how the data was collected. They argue distance measured and distance that is derived are conceptually the same context, but derived distance is not the value directly sensed. Therefore, distance as context should be disambiguated given an operational categorization. Operational categorization enables the elucidation of the important aspects and challenges of data acquisition. Cost, quality, accuracy, reliability, etc. are very important given the potential vast scale of pervasive sensing.

While Dey and Abowd did discuss the origin of context in their analysis on implicit and explicit context, it is not part of their categorization. Henricksen [116] founded her categorization on an operational scheme; specifically: sensed, static, profiled, and derived. She explored these as properties of context and their associated error. Chen and Kotz would call sensor data “raw contextual information” a form of low-level context and the context inferred from it as high-level context. They used the examples of location, noise level, and temperature as low-level and current activity as high-level [42]. Notice that under Dey and Abowd’s categorization activity would be considered a primary context. Multiple authors have considered the idea high- and low-order context, but this doesn’t address anything about accuracy and validity. Van Bunningen et al. [33] went further and developed a dichotomy of the possible categorizations of context by dividing them into: 1) Operational categorization and 2) conceptual categorization. Where operational is based on how context is acquired and conceptual is based on the relationships among context.

2.3.1. Conceptual and Operational

Perera et al. [188] proposed their own version of a categorization scheme based on Bunningen et al.’s operational and conceptual concept with respect to Dey and Abowd’s categorization. Activity, time, identity and location become the categories of the conceptual perspective, while primary and secondary are adjusted in terms of the operational perspective, Figure 2.1. Both being useful, an important distinction here is that context information can be both forms at the same time — conceptual or operational because it is point-of-view dependent.
Perera et al.’s operational context is defined as follows:

**Primary context** is any information retrieved without using existing context and without performing any kind of sensor data fusion operations.

**Secondary context** is any information that can be computed using primary context. It can be computed using sensor data fusion operation or data retrieval operations, such as web service calls.

![Figure 2.1: Perera et al.’s context categorization in two different perspectives: conceptual and operational.](image)

2.3.2. Latent Context

Presently, secondary context is a catch-all for anything that is derived from information acquired directly. This acquisition could be a database lookup, a mathematical formula, and traditional data fusion operations or classical machine learning classifiers. Modern advances in machine learning, such as deep learning, also fall into this category. These advances provide real potential to distill the hidden underlying variables of context. However, there is a clear distinction between formulas and database lookup operations as compared to the complexity and resource requirements of deep learning. Deep learning is often used in
contemporary systems, where neural networks take sensor input data and transform it using various statistical and other mathematical operations, including linear transformations. The learned representations that are found after having trained a neural network are often referred to as the “latent space.” We suggest that, information which characterizes the situation of an entity that is learned in this manner, should be called “latent context.” The need to separate the concepts of secondary context is driven not just by complexity, but also by the many reasons to have an operational categorization in the first place — distinctions in regards to accuracy, validity, reliability, cost, etc. There are clear data acquisition trade-offs which must be understood. In this work, we extend the work of Dey and Abowd, Van Bunningen et al., and Perera et al. We propose a third operational category categorized latent context.

We speculate that, in the same way word embeddings have accelerated research in the natural language processing community [160, 187], a multi-modal sensor embedding that encodes raw data and/or context into a context latent space, can enable and potentially accelerate the context-aware computing research community, this being of a more intricate nature than simple data fusion or database retrieval.

Latent variables are not directly observed, but instead are inferred from models trained on observed features. The vector space that spans these latent variables is known as a latent space. Many machine learning models map or transform observed data by projecting them into a latent vector space. This can be done for a number of reasons which include: (1) fitting a model, (2) fighting the curse of dimensionality with reduction techniques for more efficient use of computing resources, (3) data compression, (4) the generation of new data, or (5) even the reduction of model complexity, given multi-modal data. Transforming observed data into a latent vector space provides the ability to encode and preserve contextual relationships among observed data. Word embeddings are a great example of this point — dimensionality reduction is achieved and much of the semantic and syntactical relationships among words are maintained within vector sub-spaces; this is observed by their locations and distances relative to each other in the latent space.
For example, Mikolov et al. showed many analogies are preserved such as \( \text{vector(“King”)} - \text{vector(“Man”)} + \text{vector(“Woman”)} \approx \text{vector(“Queen”)} \) \cite{161}. Such inferences can enable insight into the relationships among information including measurements, text, sound, images, video, user physiological measures, etc. The situational characterization of a relevant entity can be understood much in the manner the meaning of a homonym is understood. In other words, it can translate, in a concrete manner, the distinctive nature of context data in the same way a word of many definitions is only meaningful in the context of other words.

A generative model seeks “implicitly or explicitly” to represent and approximate a latent data distribution and can be used to characterize joint statistical distributions of context information \cite{198}. A trained generative model has the potential to de-noise input or even generate new data by sampling the learned distribution, given the preserved relationships. On the other hand, discriminative models seek to understand class boundaries and learn conditional probabilities such as \( P(Y|X) \) of x input data and y labels \cite{198}. Their attempts to model class boundaries are further enhanced with the use of multi-task learning. Ruder et al. \cite{210} used multiple discriminative classifiers or tasks, attached to the same network, and trained them in parallel. Ruder showed that with multi-task learning, models tend to prefer solutions that generalize \cite{210}. In this research, the tasks we are seeking to classify are the human states we want to infer, such as cognitive load, situational awareness, and stress. Given this understanding, we redefine Perera et al.’s categorization with the following:

**Primary context** is any information retrieved without using existing context and without performing any kind of sensor data fusion operations.

**Secondary context** is any information that can be computed using primary context. It can be analytically derived from sensor data or data retrieval operations.

**Latent context** is any information that is inferred through preserved relationships among observed data given a latent space.
2.4. Conclusions

In this chapter the background of context and context-aware computing was discussed. Based on Dey and Abowd’s definition [61] we defined context as any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction among entities, including the entities themselves. We further define context-aware computing with Dey and Abowd’s definition that a system is context-aware, if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user’s task [61]. Noting the caveat that a user can be any entity. Researchers will continue to argue over such definitions. Ultimately, we concur with Mark Wiser’s vision of a computer capable of inferring context through pervasive sensors and proficient in responding, projecting, and adjusting to the dynamics of that context, all the while invisible and un-intrusive to the common awareness [275] can be realized in pervasive systems-of-systems. We see each human as a complex entity within the context-aware and ubiquitous paradigm, where a human is a system-of-systems in its own right. Therefore, we must examine the systems of the human, if we wish to infer the internal states of a human such as cognitive load, situation awareness, stress, etc. But these internal states cannot be directly measured. External physiological or biometric measurements can provide clues and act as proxies. However, ultimately latent context will need to be inferred from these physiological measures.

In this chapter, latent context was defined as any information that is inferred through preserved relationships among observed data given a latent space. In the next chapter, the techniques and methods needed to understand and learn the latent spaces needed to infer these complex measures are described. We seek to learn latent context that can help us to unravel a variety of pervasive sensor responses in order to understand these states. To do this a foundation in modern machine learning methods is required.

Finally, Rogers [204] showed that context in computing must be defined in a domain specific way in order to be useful. That is, context-aware systems are highly successful when used in constrained settings. In lieu of a domain, the general knowledge about how a system
should infer or respond is much too dynamic and variable too be operable and actionable, given the fluidity, subtleties, and idiosyncrasies of the context encircling humans as they go about their daily lives [204]. Smart computing systems struggle because they require hard artificial intelligence to solve problems that are likely as hard or harder than creating an artificial human [101,134]. Research in context-aware computing must incorporate as much domain specific knowledge and awareness of the surroundings as possible a priori. We believe the rich history and training of aviators narrows the context needed to make operable and actionable decisions about a pilot given the actions involved in flying.
Latent context was defined in the previous chapter as *any information that is inferred through preserved relationships among observed data given a latent space*. In this chapter, the techniques and methods needed to learn and understand how to infer latent context and understand the relationships that are preserved in a latent space are explored. That is, latent complex measures and relationships are inferred and preserved through deep learning. In order to do this, a background in deep learning methods is required. The ability to learn latent context may help us to uncover a variety of pervasive sensor responses, and to understand these states is key. Multiple architectures within this research take advantage of several important concepts from within machine learning and deep learning: (1) generative learning, (2) transfer learning, (3) multi-task learning, (4) multi-modal learning, (5) variational autoencoders, and (6) x-vectors. We discuss each in turn.

### 3.1. Generative and Discriminative Models

The distinction between generative and discriminative models is important to understand. Traditionally *discriminative* models seek the classify data into categories or classes. That is they learn the $p(x|y)$, the posterior probability. They attempt to estimate the probability distribution where a given observation $x$ is of class $y$. Discriminative models learn the boundary between classes. *Generative* models seek to understand the probably distribution of $x$. That is, they try to estimate the probability of the observation $x$, the prior probability $p(x)$. They can also estimate the likelihood probability, $p(x|y)$, and the joint probability, $p(x, y)$. The key difference here is that generative models predict the probability of observing sample $x$ from the learned distribution, which in-turn can generate data by sampling.
from the learned or estimated distribution [174]. In this research we used VAEs for feature extraction, and they are a generative models. We sought to classify cognitive states, which is a discriminative modeling task.

### 3.2. Transfer Learning

Leveraging prior learned weights to speed up the training of new tasks is known as transfer learning [128, 183]. These methods preserve and take advantage of previously trained models from one task or domain and apply them to a second different task or target domain. More broadly, transfer learning enables training and test among domains, tasks, and distributions to be different [183]. New accurate models can be trained through the use of these methods — models for an entirely different task and/or source domain [183]. An example can be found on Figure 3.1. For classifying gaze scan patterns we take advantage of transfer learning by borrowing the previously learned weights from the Visual Geometry Group’s 16-layer model [234].

![Figure 3.1: Transfer Learning Example](image-url)
3.3. Multi-task Learning

Another key concept is the preservation of learned knowledge while training multiple tasks; this is known as multi-task learning. Multi-task learning traditionally differs for knowledge transfer because the shared knowledge is learned at the same time, among tasks, and throughout training. However, multi-task learning can be considered a form of transfer learning.

A common goal for multi-task learning is to unveil the shared latent features that benefit each task [183]. Further, multi-task learning has an innate ability to generalize. This tendency or model preference towards generalized solutions is explored and validated by Ruder, [210]. One key avenue of this tendency is to use auxiliary tasks to improve the performance of the model. Ruder pointed out that multi-task learning is a logical fit for learning desired tasks at once. It is often the case that only one task is desired and tasks considered to be “throw away” are chosen in order to boost the performance of the model. The selection of suitable tasks is a research problem in and of itself. However, a common method is to consider “related tasks.” Our deep learning architectures, Section 8.3, use related tasks to improve performance; specifically, we use binary, multi-class categorical, and regression. Multi-task methods can also attempt to combine unrelated tasks together, sometimes referred to as multi-domain processing. In our work, for example, we might refer to a model that classifies Bedford workload as a different domain than classifying mental demand from the NASA-TLX. However, within the domain of Bedford workload, we can refer to related tasks (such as binary and 3-class categorical) as multi-task processing.

3.4. Multi-modal Learning

Lahat et al. noted that given the observation of phenomena using multiple sensors, this sensor output can be termed a modality, where it is associated with a single dataset [142]. The use of multiple modalities is motivated in taking advantage of complementary information within modality type. There is potential to produce a richer feature set that can yield better performance when compared to a single modality [198] because different modalities can
capture alternative, but complementary information from different vantage points. A key issue within multi-modal learning is where to merge or fuse the different modality features.

There are three key types of multi-modal merging or fusion: (1) early or data-level fusion, (2) late fusion, and (3) intermediate fusion. Early fusion consists of first merging different modalities, either raw or pre-processed, prior to input into a model. Late fusion is also known as decision level fusion because it is the “aggregation of decisions from multiple classifiers” [198] where each classifier is trained on a single modality. There are many ways to merge multiple modalities, including maximum, averaged, Bayesian based, etc. [198]. Ensemble classifiers such as random forests are a great example of late fusion. Intermediate fusion occurs when deep neural networks transform raw features into latent hierarchical representations through many linear and nonlinear transformations given multiple deep layers [198].

Ramachandram et al. pointed out that a key method in deep learning is to learn latent hierarchical representations given raw data, e.g., as with a variational autoencoder (VAE) or with X-vectors. These feature extraction techniques can enable “fine-grained” control of how the modalities are merged within deep, multi-modal learning architectures. Specifically, they note the “common practice” constructing a merge layer that forces the model to learn shared representations of merged input data streams [198]. The most common form being a layer of hidden nodes receiving input features from all modalities. There is a level of flexibility with learning cross-modality, shared representations within different levels of abstraction [198]. Common design choices include: when to fuse? what to fuse? and how to deal with modality loss? Our $BM^3TX$ architectures, Section 8.3, uses the intermediate approach. Through VAEs, Section 3.5, and X-Vectors, Section 3.6, we learn these higher level representations.
3.5. Variational Autoencoder

Variational auto-encoding, a generative method, was used to reduce dimensionality and facilitate feature extraction. Variational autoencoders (VAE) with convolutional layers were used to “encode” our 1-D modality data streams. This method of encoding is trained using reconstruction error and KL-divergence, where it is assumed the actual posterior is an approximate multivariate Gaussian normal distribution with a diagonal covariance; and the prior with respect to the latent variables are centered isotropic multivariate Gaussian $p_\theta(z) = \mathcal{N}(z; 0, I)$ [136]. The parameters $\mu$ and $\sigma$ are outputs of the encoder, where it is assumed the approximate posterior is log $q_\phi(z|x^{(i)}) = \log \mathcal{N}(z; \mu^{(i)}, \sigma^2(i), I)$. KL-divergence incentivizes the approximate posterior to learn a representation similar to the prior, and yields a low dimensional latent space with an approximate diagonal covariance, which is then used in reconstruction to decode and approximate the original signal.

With other auto-encoding methods the learned latent space can be erratic and non-continuous [136]. In order to mitigate this in VAEs, the mean and variance are used for sampling vectors from the approximated prior, where this prior has been re-parameterized in the form of $z^{(i)} = \mu^{(i)} + \sigma^{(i)} \epsilon$. Here $\epsilon$ is an auxiliary noise variable, a sample from a zero mean, isotropic multivariate Gaussian [136]. This auxiliary noise of variational auto-encoding has been shown to force continuous relationships in the encoded space that are easier for machine learning algorithms to interpret. In this way, we can find descriptors of the time series data without the need for labeled data. Once the low dimensional encoding generates acceptable reconstruction loss, the reconstruction phase of the network, the decoder is eliminated, and we use the encoded latent space for prediction. We note that an alternative to the use of VAEs is an adversarial auto-encoder, which has been shown to impart desirable properties into the latent space [152]. However, these methods are notoriously difficult to train consistently because of the adversarial loss landscape employed; and therefore we adopt the VAE methodology.
3.6. **X-Vector Inspiration**

Our architecture borrowed key concepts from the x-vector training architecture commonly employed in speaker verification systems [239]. X-vectors are fixed-dimensional embeddings trained, given a deep neural network, on variable-length speech utterances. They are instrumental within the domain of speaker recognition as they create embeddings to capture speaker characteristics over the entire utterance and generalize well to other speakers [239, 240]. More specifically, the core architecture uses time-delay neural networks at its frame-level layers. The features learned are aggregated within a statistics pooling layer, where mean and standard deviation are taken across the temporal dimension of final frame-level output. At the segment-level, this pooling layer’s output is passed to two additional dense hidden layers before reaching the final softmax output layer. The dense learned features at either dense layer are then extracted and used for speaker recognition [239]. The key aspect of this architecture is its ability to learn latent features which capture the speaker characteristics from variable-length utterances. While our architectures are end-to-end, we borrowed key aspects from the x-vector architecture, including (1) the statistical pooling layer that enables ‘segment-level’ classification (in our case, across the entire observation), and (2) the hierarchical learning of latent features, where each dimension is an abstraction of the input observation. While they utilize time-delay neural networks, we used more traditional 1-D convolutions and separable convolutions.
3.7. Conclusions

In this chapter, key deep learning methods needed to infer and understand the relationships that are preserved through learning a latent space are discussed. With the capacity to learn latent context, we can unveil a variety of pervasive sensor responses and relationships in order understand the human-in-the-loop. The important concepts included: (1) generative learning, (2) transfer learning, (3) multi-task learning, (4) multi-modal learning, (5) variational autoencoders, and (6) x-vectors. In order to use these methods to build the models necessary for an inference system capable of assessing human cognition, we must understand key applications and prior work. Therefore, we turn to the discussion of situation awareness, cognitive load, and learning.
Previous research in cognitive load for aviators has established a relationship between cognitive workload and various student performances. However, the quantification of that relationship has not been amply researched. That is, the diagnostic power of physiological measures to personalize learning for aviators is still unrealized. We seek to establish the degree to which cognitive workload and task performance co-vary in the domain of aviation using physiological sensors as indirect, real-time indicators. Further, it is essential to understand how these indicators change between training scenario and subject. We hypothesize that context-awareness for aviators is both agnostic to scenario and subject, and is extensible to a number of training scenarios of sufficient similarity in terms of duration and cognitive complexity.

When training, the performance of aviators is currently measured through instructor observation and review. Through this observation instructors can intuit the internal context of aviators while they learn, which is important for assessing their development of many skills. While understanding this internal context is recognized as important, the automated sensing of this context has not fully materialized. By inferring learner context automatically, instructors and trainees can more easily determine when mastery of a particular skill is achieved, expediting the learning process, and potentially, improving the overall quality of learner understanding.

Society may benefit from a mechanism to more safely and quickly develop skilled aviators. At present, the aviation industry, both civil and military, cannot produce competent aviators fast enough, driving up financial costs and delays, while reducing safety levels. Understanding the internal context of an aviator during a simulation are important aspects of the learning experience. It may be possible to personalize the learning experience for student
aviators, and expediting the educational effectiveness by maximally challenging the learner without overloading them. To this end the previous work in several disciplines must be understood in order to build the models necessary for an inference system capable of assessing learning. They include: (1) situation awareness, (2) cognitive load, (3) cognition-aware computing, (4) gaze, and (5) personalized learning.

4.1. Situation Awareness

A well-established cognitive construct delineating the state of knowledge about a dynamic environment is known as situation or situational awareness (SA). A pilot’s SA is highly correlated with pilot performance [79]. The importance in maintaining accurate SA cannot be understated. It is crucial for ensuring both safety and mission success. Often it has been the case that lack of SA was directly associated with pilot error [81,90]. Therefore, the need to accurately measure SA is important in all areas of flight and including the improvement of training design and overall training results [80]. Moreover, Hart, one of the authors of the NASA Task Load Index subjective workload (NASA-TLX) scale, found situation awareness to be one, if not the most popular co-variates in research regarding NASA-TLX and various factors with respect to successful performance. Further, it was suggested that “SA is simply a consequence of workload and not an independent phenomenon” [112,114]. Conversely, Endsley’s work in SA provided validation as a higher-order cognitive construct [184]. Parasuraman et al., cited the need for viable constructs in the promotion of elucidating the human variable in complex systems, “show that the constructs can be operationalized using behavioral, physiological, and subjective measures, supplemented by computational modeling, but that the constructs are also distinct” [184]. They found not only SA, but also mental workload to be significant to “the understanding and prediction” of human-system performance.

Situation awareness global assessment technique (SAGAT) is an exercise used to evaluate the SA of a subject by comparing subject understanding of a situation against reality; this information is based on the subject’s perception and not inferred from observation which
can contain bias. A test subject’s perception of the present situation is collected during the scenario, reducing error caused by collecting information after the fact. This is accomplished by randomly freezing the scenario and blocking out the screens. At this point the subject would be asked specific non-subjective questions about the current situation. Multiple snapshots of subject’s SA can be collected, providing an index of learner’s performance enabling an objective assessment [80].

SA in aviation, especially in the training of new pilots and other aviators, occupies a pivotal role. Much effort has been conducted towards the understanding of pilot physiology, eye movements, and scan patterns, which “shed considerable light on [aviators’] real-time behavior” in aviation and aerospace as a whole [263]. Flight instructors seek to comprehend the intent and awareness of their students; and this context is currently derived from in-flight observation and post-flight examination. Because of this, situation-aware avionics — capable of assisting either the flight instructor or the aviator in understanding the human-machine state [36] in real-time — is highly sought after. Key motivations include accelerating aircrew training, improving aircrew performance, and decreasing pilot workload. Our research could potentially provide a form of context or situation awareness for use in a broader evaluation system.

4.2. Cognitive Load and Workload

There are mixed views on what cognitive load or workload is — and these two terms are often used interchangeably. In the context of aviation, Cooper and Harper defined workload as “the amount of effort and attention, both physical and mental, that a pilot must provide to attain a given level of performance” [53]. Where physical workload is described as the “effort expended by the pilot in moving or imposing forces on the controls during a specified piloting task” [53]. Mental workload, being much harder to quantify and separate, was described in the amount of mental compensation required by a pilot to complete a task [53]. Roscoe and Ellis found that 80% of military and commercial pilots view workload “in terms of effort.” They found that when pilots rate the amount of workload, they assess it from the perspective
of “spare capacity.” Therefore, they redefined workload as “the integrated physical and mental effort generated by the perceived demands of a specified piloting task” [208]. Their Bedford Workload scale emphasizes the available human resources as it pertains to workload. The creators of NASA Task Load Index (NASA-TLX), Hart and Staveland, defined workload in terms of “a hypothetical construct that represents the cost of accomplishing mission requirements for the human operator” [112, 113]. In Hart’s paper, titled NASA TLX; 20 years later, she pointed out a prevailing theme with workload in the psychological literature. Workload in literature has many interpretations, and it is a “testament to the complexity of the construct” [112]. One such definition: cognitive load can be described as the load that a task or activity places on an individual’s working memory [17, 43, 130]. Therefore, given the disagreements amongst subject matter experts, it is no surprise that test subjects would have similar and yet diverse thoughts and opinions about how they perceive workload [112]. Thus, in attempting to build a classifier that works across subjects, this can present an obstacle. In this section we look at cognitive load in cognitive science and the associated cognitive load theory (CLT) with respect to learning. We delve into methods for measuring cognitive load. Finally, we explore a method for inducing high workload in flight — Boundary Avoidance Tracking.

4.2.1. Cognitive Science and Working Memory

Cognitive load can be viewed as a measure of the task on the available resources given an individual’s working memory. As with cognitive load, working memory has varying definitions of its own. Still, it can be thought of as a cognitive system, centered around short-term memory and the mental processes involved — “holding information in [the] mind and mentally working with it” [66]. Baddeley et al. described working memory as “multiple specialized components of cognition that allow humans to comprehend and mentally represent their immediate environment, to retain information about their immediate past experience, to support the acquisition of new knowledge, to solve problems, and formulate, relate, and act on current goals” [14]. Fuster described working memory in the context of
neuroscience and phenomenology as “sustained attention focused on an executive cognit for the processing of prospective action” [91]. That is, working memory can be thought of as “attention focused on an internal representation” [91]. The limitations of working memory have been heavily explored [7, 13, 17, 40, 52, 55, 130]. The idea of a limited working memory can be traced back to George A. Miller [162], who suggested that, in general, the human processing capability is limited to near seven elements. That is, humans can hold around seven numbers in their immediate memory before loss. It has been shown that working memory capacity can refer to either short-term memory limitations, processing power, or both [48, 51, 55, 140, 176, 177, 216, 257] — therefore, it is more apt to refer to memory capacity as the mental bandwidth of working memory.

4.2.2. Cognitive Load Theory

Cognitive load theory (CLT), [182, 251, 252] is an area of cognitive science that focuses on the “development of instructional methods that efficiently use people’s limited cognitive processing capacity to stimulate their abilities to apply acquired knowledge and skills to new situations (i.e. transfer).” CLT works with a cognitive architecture that encompasses limited working memory and semi-independent processing units for visual and auditory information, which interacts with a much larger, long-term memory [179]. Research in CLT has focused on the use of innovative instructional methods to efficiently use working memory capacity [179]. In this sub-discipline of cognitive science, cognitive load is a construct defined formally as follows:

The cognitive load construct is multidimensional and it represents the load that performing a particular task imposes on the learner’s cognitive system. [182]

Paas et al. describe two dimensions or factors, “factors that affect cognitive load and factors are affected by cognitive load” [182]. These include causal and measurement where the casual dimension includes the characteristics between the learner and the task, such as task complexity and time pressure. The measurement dimension includes key concepts: (1) mental load, (2) mental effort, (3) performance. Mental load is an indicator for expected
cognitive capacity demands. It is based on the current understanding about the task and subject characteristics including environmental aspects. It is considered a priori for cognitive load [179]. Mental effort is an indicator of expected cognitive capacity allocated to support the demands driven by the task, and an echo of how much cognitive workload the learner has when conducting the task. This is the amount of effort expended by the learner, and it is considered a reliable approximation for cognitive load [179, 182]. Finally, performance is evaluated as the learner’s achievements such as correct items, task duration, number of errors made, etc. [179]. It is important to note that within this cognitive load construct, it is possible to have two subjects achieve the same task performance while expending discernibly different amounts of mental effort [179].

CLT further categorizes cognitive load into three types of load: (1) intrinsic load, (2) extraneous load, (3) germane or effective load [179, 182, 252].

1. **Intrinsic load** is driven by the innate nature or complexity of the task being conducted and the related knowledge of the subject conducting the task. It is generally considered unalterable.

2. **Extraneous load** is artificial because it is imposed by the presentation and may include inefficient processes or methods.

3. **Germane load** is the mental effort that contributes to the development, construction, and automation of schemas.

Long-term memory is the store of a person’s knowledge [45]. The form this takes is modeled as a schema. A schema categorizes elements of information in the manner in which they are utilized [45]. As new information is learned, schemas are constructed and developed through modification. However, information must be “extracted and manipulated in working memory” before it can be stored in schematic form [179]. Further, schemas reduce working memory load by organizing information in such a way as to treat many informational elements as a single element [252]. Working memory load is further reduced through the automation of schemas.
Information is processed through conscious or automatic means [233]. “Conscious processing occurs in working memory, [where as] automatic processing largely by-passes working memory” [252]. This can be demonstrated when reading as most people do not “consciously process...the individual letters that make up the [words] being read” [252].

![Figure 4.1: Attributes of cognitive load][1]

Xie and Salvendy [279] described several parameters of cognitive load given the construct. Specifically, they described instantaneous, peak, accumulated, average, and overall load, Figure 4.1. Instantaneous load is the current value of cognitive load at any given time. Peak is the maximum value of instantaneous load that occurred during a task. Accumulated load is the total amount of load over the entire task, the area under the curve. Average load is the mean value of instantaneous load during the whole task. The overall load is a task and subject dependent mapping of instantaneous workload or accumulated and average workload to the subject’s brain.

---

[1]: 37
Instructional design that minimizes extraneous load increases the unused working memory. Instructional encouragement that maximizes germane load leads to the development, construction, and automation of schemas. The “chunking” of information into “single elements of cognitive schemas” and further automation enable some circumvention of working memory limits. The “essence” of CLT is derived through the use of “instructional control of cognitive load to attain transfer,” [179]. It is the transfer of known skills and information towards new situations — human knowledge transfer. The key to this end and this research is the ability to measure cognitive load.

4.2.3. Measures of Cognitive Load

There are two key ways to measure or approximate cognitive load, subjectively and objectively. Subjective measures often vary. Because subjective measures are rating scales, they are based on the assumption that people can, to a certain degree, reliably self-evaluate and report the mental effort required to complete a task, [179]. Pass et al. noted that this has been demonstrated on several occasions [43, 112, 113, 180]. Moreover, subjective rating methods used most often target overall workload, Figure 4.1, as these techniques usually involve a questionnaire that is delineated by one or more semantically separable scales. In this research, we do the same. Pass et al. further noted that while most are multi-dimensional, using variables such as “mental effort, fatigue, and frustration” [179], single dimension scales have been shown to be sensitive, valid, and reliable [92, 182]. We utilize both the NASA-TLX (multi-dimensional) and the Bedford Workload (uni-dimensional) scales to measure the overall workload.

Before continuing, it may be helpful to provide a few notes about the NASA-TLX. It is comprised of six sub-scales: (1) mental, (2) physical, (3) temporal demand, (4) frustration, (5) effort, and (6) performance — Appendix B.1. NASA-TLX defined a weighting scheme that can be applied, which was intended as a method to reduce subject variability and increase sensitivity. A common modification is to not use this weighting scheme at all — calling it Raw TLX (RTLX). Interestingly, there have been mixed results with the use of
the weighting scheme, showing evidence the that scheme works [115], or that it doesn’t work [149], or that it has no effect [23]. In our research, we have chosen to use RTLX for both time constraints and to help reduce the intrusion on the ongoing experiment. Finally, researchers and subjects continuously point out the sub-scales are often highly correlated with one another [112], which we have found to be true in our experiments. Hart pointed to the idea that this is an illustration that the sub-scales are all evaluating some aspect of the same underlying latent entity [112].

**Objective measures** can also vary, and they can be evaluated in two fundamental ways: (1) through the use of performance data with primary and potentially secondary tasks and (2) through the use of physiological or biometric measures. Performance-based measures come in two flavors: (1) primary task only — based on primary task performance; or (2) primary and secondary tasks — evaluation is concerned with the performance of the secondary task while conducting the primary task. In other words, the evaluation is looking at the “spare capacity” to conduct the secondary task, given the primary task. Typically, the secondary task is simple in nature, requiring sustained attention, but it differs from the primary task [179]. In our feasibility study, we looked at a secondary task as a way to drive up the workload, Section 6.2, but ultimately deciding against it. As Pass et al. observed, “it can interfere considerably with the primary task” [179] and, in our case, it is the biometric data we are trying to collect and from which we can infer cognitive load. That is, certain biometric measures would sync in time with the pilot attending to the secondary task, which is not a realistic observation of what we would expect from the biometric data in a normal flight environment.

Physiological or biometric measures are based on the idea that evidence of cognitive or mental function is present in psychological or biometric responses [179] — measures like heart rate variability (HRV), pupil dilation, and galvanic skin response. These physiological markers are then evaluated as a measure of cognitive load [21, 35, 109, 124, 181, 190, 258, 277]. Of note, Ursin and Ursin observed that such physiological measures do not measure “imposed load,” but instead, they inform how the subject themselves perceives load, and how they “cope with” such workload [258], which is ideally what we want to capture [109]. Indeed, it is
possible to have two subjects achieve the same task performance while expending discernibly
different amounts of mental effort [179]. The use of physiological measures for classification
of cognitive load is a key goal of this research. Given the previous research discussed here,
we can now properly formulate the following research question:

*Can we accurately and reliably use multiple physiological/biometric modalities to ob-
jectively evaluate cognitive load as latent context across subjects and across tasks?*

4.2.4. Boundary Avoidance Tracking

In order to classify cognitive load, maneuvers of varying mental load (characteristic) are
required to induce multiple levels of load in test subjects while flying. Anecdotally, most
any pilot will tell you the amount of load on a pilot in a simulator compared to actual flight
differs significantly. There are some things simulators cannot replicate — for example, the
response given the fear of hitting the ground when landing. In order to induce pilot workload
in a simulator, full engagement and buy-in are required. Therefore, we induced high levels
of workload through the use of Boundary Avoidance Tracking (BAT) theory [100], which
originates from the flight test community.

A flight test technique (FTT) is a methodology founded in engineering principles and
used in flight test to determine the characteristics of an aircraft. The test data gleaned from
these techniques are typically used when evaluating an aircraft to ensure it meets a specific
pre-existing requirement [100]. Boundary avoidance tracking is a flight test technique used to
understand the “pilot-in-the-loop handling qualities” [100] such as pilot-induced oscillation
(PIO). PIO occurs when a pilot drives the “pilot/aircraft system into instability,” which can
manifest itself when the pilot input is out-of-phase with the aircraft [98,100]. BAT is founded
in the concept of avoiding boundaries; this occurs when a “pilot controls an aircraft to avoid
a condition rather than maintain a condition” [99]. Examples of boundaries include pitch,
G limits, the ground, position with respect to a wingman, etc. The key here is that the pilot
is tracking the aircraft state in order to avoid a condition, not maintain one. By reducing
the boundaries in a buildup manner, heavy workload can be induced as a consequence.
Gray showed that as the boundaries collapse, the allowable error decreases, and actual error decreases with increasing performance. However, this only happens to an extent. Once the allowable error is sufficiently small, instabilities occur, and “workload...increase[s] until the pilot can no longer accomplish the task” [100]. Figure 4.2 visualizes the progression as an allowable error decreases. Gray likened this to riding a bike across a beam or an elevated narrow path at an unsafe height. If this path is progressively made “narrower and narrower” further along the path, the “rider will clearly increase his effort to stay” in the center of that path. Once this path becomes narrow enough the rider can no longer remain stable and “oscillates off the” path [100].

The BAT workload buildup FTT is defined as a process where the tolerance for error or boundary is reduced in a step-wise manner — driving up the difficulty with each step. The pilots are expected to ‘role-play’ as if the boundaries are safety-critical. There is a final step called the ‘switch-induced simulated PIO’ which is designed to force the pilot/aircraft system into the instability of a PIO, which is a handling qualities evaluation and is not needed for this research, as we are interested in the consequence of the BAT FTT the induction of workload. In our studies, we have sought to drive maximal load. If a test subject found themselves in a PIO, prior to the maneuver we asked them to stay in it as long as possible until they can no longer fly the maneuver as this is a high cognitive load state. The boundary avoidance tracking task FTT was paramount to increasing the workload during the collection of data needed in order to build a workload classifier.
4.3. Cognition-Aware Computing

Cognition-aware computing is a research area that primarily focuses on the use of cognitive models with user-facing systems [31, 32, 138, 194–196]. It is the application of context-aware computing [61] by inferring the latent cognitive states or “cognitive context” [31] of the human user and doing something with that context (for example, changing a user interface). Measures of cognitive context can include cognitive load, stress, attention, engagement, affective state, etc. Many works in this research area focus on finding operational measures that are ‘good enough’ for use in a computing system. Highlighting key uses of cognition-ware systems, Rafiqi et al. [195] noted adaptive user interfaces with applicability in the security and safety critical medical fields are important applications. Bulling and Zander [32] underscored key applications such as the dynamic adaptation of game demands based on engagement [173], memory-recall assessment [30], or “supporting people during safety-critical tasks by monitoring cognitive workload and fatigue” [32]. Safety-critical applications can include, among others, assisting a doctor in critical surgery, an automobile driver, or in the case of this research an aviator flying an airplane — all key safety critical tasks that can
benefit from understanding “cognitive context” [31]. Thus, there is an expansive need to accurately measure or evaluate cognitive load. Bulling and Zander aptly pointed out high value in reliably measuring induced cognitive context in a controlled environment, where it is then “robustly inferred” in life’s daily complexities [32]. Indeed, these ideas predate the Pervasive and UbiComp communities themselves. Researchers in the Aerospace, Psychology, and HCI communities sought to deliver physiological measures for (1) subject screening, and (2) system development across all phases including final system evaluation — “to discern if a workload problem exists”; and (3) “on-line monitoring for workload” where moment-to-moment workload is determined and applied to automated systems designed to “reduce the load” [108, 277]. This final, real-time monitoring characteristic was often considered “most valuable” during these early works; however, the complexity of such a system has prevented it from becoming a reality even in contemporary computing environments.

Recent works in cognition-aware computing highlight the potential of real-time monitoring. In Haapalainen et al.’s seminal work [104] they collected data from 20 subjects using multiple physiological modalities and NASA-TLX subjective ratings. They modeled two levels of cognitive load using only short common tasks found within UbiComp community and situations of divided attention — tasks that target visual perception and cognitive speed. The electrocardiogram median absolute deviation and median heat flux (rate of heat exchange) measurements were most useful, attaining an accuracy of 81.1% intra-subject and intra-task. Unfortunately, while the same modalities produced the best results across all subjects, they were unable to “build a [singular] model with high classification accuracy.” That is, the system required considerable calibration data for each subject to be accurate. This disadvantage is often present in systems built to be cognition-aware. As such, the contributions of such systems are often limited by their scalability.

Chen et al. compared four methods for measuring cognitive load across 15 subjects, they included: (1) subjective ratings, (2) task completion time, (3) performance accuracy, and (4) physiological measurements (pupillometry and blink rate) [43]. They found subjective ratings are the most accurate (intra-subject) as compared to the mental load characteristic [190] of
the task — task difficulty. While different subjects may have different levels of load for the same task, Chen et al. showed the discriminative power for within-subject load classification, given the subjective ratings (through repeated ANOVA). Using the induced mental workload task levels, i.e., the mental load characteristic, as ground truth they built subject-dependent Gaussian mixture models and built binary intra-subject models based on subjective ratings with an accuracy of 90%.

Abdelrahman et al. found they could unobtrusively detect changes in cognitive load within a subject during a Stroop Test [250] using thermal imaging. They evaluated 25 subjects, detecting temperature changes near the nose and forehead with an average latency of .7 seconds after stimulus covering four levels of load within a subject [4].

In previous work using smartphones to infer pupillary response, [194, 270] updated the pupillary sensing system with a calibration protocol that brings pupillary responses of a diverse range of people and lighting conditions onto a single 0.0-1.0 scale called CogPoint. The new calibration required users to stare at the screen of the phone briefly, and a randomized parameter search then selected values for the individual and for the lighting with regard to the filtering and segmentation of the iris. Furthermore, they updated and optimized the algorithms employed to run in real-time from a smartphone. While all mentioned works are significant to the understanding of cognitive load in a computing environment, none of these works were successful in building a classifier of cognitive load that works across users and across different tasks.

An important research goal has been to develop a model that classifies workload across subjects (an inter-subject workload model). Towards this goal, Appel et al. used six-second, eye-related, philological measures, including pupillometry and blink rate, and a modified n-back task [219] to develop a set of classifiers capable of a binary, online, inter-subject accuracy of 70.4%. In their work they employed “a weighted voting system to detect workload for an unknown subject” [8]. However, they built their models to classify the mental load characteristic, an issue they note in their paper. Mental load characteristic is “an a priori estimate of...cognitive load” stemming from the relationship of task and subject providing

44
a presumed indication of “cognitive capacity demands” [179]. In this case, the three levels of difficulty of their n-back test is the mental load characteristic, and not the level of load the subject was experiencing or “coping with” [258], Section 4.2.3. They found accuracies of 53.8% (0 vs 1), 70.4% (0 vs 2), 66.8% (1 vs 2), and 37.8% (0 vs 1 vs 2). Unfortunately, this inconsistency makes sense, and is a testament to the difficulty of the problem of generalization. As Chen et al. have shown, the best discriminator of subject experienced workload is the subjective rating, which is only understood intra-subject [43,112]. Further complicating matters is that this “best discriminator” is in and of itself noisy [112]. In our research, we attempted to mitigate these effects by using the NASA-TLX and Bedford workload scales (both intra-subject scales), with a novel subjective label alignment scheme (Section 8.2.1). Once labels were properly aligned, we developed an inter-subject classifier. While we do not claim that the problem of generalization has been solved, our methods mitigated the effects of these noise sources considerably, within the context of flight.

4.4. Cognitive Load and Personalization Systems in Aviation

Previous research in cognitive load for aviators and operators has established a relationship among cognitive workload and various performances. In the late 1980s a study was conducted using electroencephalogram (EEG), heart rate (HR), and eye blinks of pilots flying 90-minute missions, a “four ship formation” [236]. Each pilot flew the same mission in both actual flight and in the simulator. They correlated the more difficult portion of the mission with higher HR, fewer eye blinks, and increased EEG activity for both the simulator and the aircraft, an A7. This same pattern of correlation was found irrespective of wing position (pilot lead vs wing-man positions within the formation). However, the amplitude of this pattern was reduced within the simulator, as the lowest HRs and highest blinks were recorded. “These data [validates] the fact that physiological measures are useful indices of pilot workload in actual flight conditions, and can be used to compare flight vs simulator missions” [277].
During the first decade of the twenty-first century, cognitive load in personalization systems for operators and aviators had been investigated through a number of efforts. DARPA’s vision for the AugCog program was to develop a “robust, noninvasive, real-time, cognitive state detection technology for measuring the cognitive processing state of the user” [224]. And through the Integrated Intelligent Flight Deck (IIFD) Project, NASA promoted a “future flight deck system [that] is aware of the vehicle, operator, and airspace system state.” They envisioned a system that could sense and adapt to “internal and external hazards...providing key information to facilitate timely and appropriate responses” [170]. Unfortunately, at this time the equipment was considered too bulky, and much of the processing was conducted offline or post-task. For modern hardware, new low-cost, wearable devices and the pervasiveness of hardware-accelerated machine learning systems on mobile devices have mitigated these concerns. While many research experiments have been carried out showing the ideas to be promising, none of this research has resulted in a truly robust implementation.

Salden et al. [215] validated the hypothesis that dynamic task selection leads to more efficient training. They evaluated the differential effects of four-task selection methods on training of air traffic control trainees. They reviewed one non-dynamic condition where learning tasks were presented in a fixed order, as well as three dynamic conditions where presentation order was based on performance, mental effort, or both, showing that dynamic task selection leads to more efficient training. While air traffic control are not aircrew, they are excellent proxies given their level of task saturation and position in the aviation domain.

A number of studies utilized electroencephalogram (EEG) and eye tracking sensors [68, 75, 146, 227]. However, other sensors have been evaluated including speech prosody [121], functional near-infrared spectroscopy (fNIRS) [111], electrocardiogram (ECG), galvanic skin response (GSR), heart rate, and respiration [225], pulse oximetry [228], and thermal Imaging [248]. The results of these works were mixed, as most works are only proof of concept, collecting preliminary physiological data and showing that the sensor data is high fidelity, even in the context of flight. A number of studies made broad, non-specific conclusions about
the physiological signals such as there exist[s] significant differences in the extracted features among different subjects [146]. Operator Performance Laboratory (OPL) has published several papers on the successes of specific sensors [75, 146, 225, 228]. Their sensor suite consisted of eye and head tracking, ECG, EEG, Electromyography (EMG), GSR, respiration, oxygen saturation ($SpO_2$), facial thermography, pulse wave, and fNIRS. In [226], they sought to assess workload with a tool that works to unify flight data with physiological measures into a single framework, in flight and in real-time. They released a demonstration paper about their capabilities under a training task example [228].

More recently, OPL participated in the *Objective Measures of Pilot Workload* study [154] at Edwards AFB. Using data collected from seven test pilots over five real world flights in a C-17 military cargo aircraft and a simulator, Martin et al. found they could measure intra-subject relative workload using an ECG sensor that appears to follow the trends of workload. They use an algorithm they call a chaotic physiological classifier, the foundation for which is based in chaos theory [82, 154]. Because this was a workload measure relative to the subject, this metric needed to be normalized for each pilot’s physiological response. During flight they used ‘organic evolution’ where they scaled the workload metric by taking “the minimum and the maximum observed values of” a subset of a pilot’s maneuvers defined by high and low workload a priori “into a 1..10 range.” Given the limited data set, only so many conclusions can be made.

### 4.5. Adaptive or Personalized Learning

Cognitive load research has shown, [21, 43, 124, 179, 208] the strong potential in eye tracking and pupillometric measures for inferring cognitive load. In the context of adaptive or personalized learning, there have been multiple works that utilize cognitive load monitoring. Porta et al. described the use of eye-tracking to measure cognitive workload when teaching mathematics to students in the e-learning environment [191]. They pointed out the inherent advantage of face-to-face learning over non-adaptive digital learning. Human instructors can understand and infer the signs of cognitive effort, stress, or tiredness. In their research,
Porta et al. remotely tracked pupillary dilation with an infrared eye tracker. Their results revealed the amount of pupillary dilation correlates to the amount of mental effort required to solve math questions. Similar studies have further shown that eye tracking can contribute to understanding users in the e-learning environment [38,41,169,175]. Concerning eye tracking, a plethora of modalities beyond pupillary dilation monitoring have been researched, including: (1) the use of blink speed and (2) blink frequency [22], (3) saccade velocity, (4) duration, and (5) amplitude [63], and (6) micro-saccades (small saccades around attempted fixations [64,65].

Beyond eye-tracking, Muth et al. [167] investigated respiratory sinus arrhythmia, and Hoover et al. [118] proposed a new approach to using heart rate variability to measure changes in real-time workload. Seminal research by Arroyo et al. [9,11] presented an adaptive tutoring system, MathSpring, that dynamically changes question difficulty given the number of incorrect answers and hints that a student has requested. The adaptive mechanisms use the concept of effort-based tutoring [10]. Adaptive-based tutoring strives to model learners and item difficulty. Through the use of decision rules, it endeavors to keep students in the zone of proximal development, which is a model first introduced by Vygotsky [266]. Murray and Arroyo [166] utilized the zone of proximal development as a balance between learner effort, given previous questions, and the difficulty of upcoming questions. With this adaptive system, school-aged children have benefited with an average increase of 10% on standardized mathematics exams. Currently, MathSpring only covers mathematics problems.

Further, Arroyo et al. have recognized the need for adding additional sensing modalities to understand context about the learner fully. They had begun an investigation on integrating sensing with MathSpring only to stop shortly into their pilot study. Arroyo et al. found, ...sensor-free affect detection is a more scalable method with larger numbers of students [11]. Therefore, the relevance of cognitive load monitoring and physiological measures in adaptive or personalized learning is recognized but has been unrealized. Therefore a research question remains. Can the use of physiological measures for personalized learning be demonstrated, and to what degree of acceleration can be provided?
4.6. Conclusions

With an insight into the constructs for cognitive load and situation awareness, and by exploring the work that has been put forth in the areas of aviation, adaptive systems and learning, we can begin to understand the data we would need for developing a model. Despite the broad research completed in the areas of cognition and aviation, we must ultimately conclude that no one has succeeded at a statistically sensitive, task-agnostic, and subject-agnostic objective pilot workload classifier. To capture cognitive load with deep learning, we need to induce multiple levels of workload that we can label; this includes both extremes: minimum and maximum workload. For low load this can be done with a fundamental cruise maneuver. And for high workload, the boundary avoidance tracking theory provides the means drive up cognitive workload in an aviator, even in a simulator. We need to understand and evaluate the capabilities of the available sensors and the physiology that they can capture. In the next chapter we explore the feasibility of a sensor suite that we utilized in a simulator and in actual flight.
Chapter 5
MODALITIES

In this research, we use a number of sensing modalities including photoplethysmography (PPG), electrodermal activity (EDA), peripheral skin temperature, and wrist acceleration in conjunction with gaze metrics including pupillary response and eye blinks. We hypothesize that combining these modalities could help provide insights into the cognitive load of an individual more reliably than previous works. In order to keep the form factor for the biometric measurements less intrusive, mobile, and compact, an all-in-one, wrist-worn device, we selected the Empatica E4 wrist band. The device was issued with FDA approval for monitoring seizures, but the research-based device is simply used as a data collection instrument. We also used the HTC VIVE Pro Eye, which contains an integrated Tobii eye tracker that provides eye-tracking and pupillary response measurements.

5.1. Photoplethysmography

Photoplethysmography is a plethysmograph which employs specialized optical techniques, such as optical reflection. The E4 uses a pulse oximeter to accomplish these techniques and measures oxygen saturation level ($SpO_2$) in the blood. The E4 calculates the change in the arterial blood volume where the systolic point and diastolic points of a heart beat are the most and least oxygen saturated. This change in measured oxygen saturation generates an output signal known as the pulse wave [76, 235, 256].

A number of measurements can be inferred directly from a continuous PPG signal, including heart rate, heart rate variability (HRV), and inter-beat interval (IBI). Kahneman showed that heart rate can be used for precise tracking of mental effort during performance of a mental task [131]. Further, heart rate in aviators has been shown to increase with levels of workload while in flight [206, 207, 237] and in simulators [147, 148, 178]. The original
relationship between cognitive load and HRV was established by Lacy [141] and validated by Matrin and Graham et al [97,153]. As task difficulty increases, suppression of heart rate variability is observed [2, 109, 132, 165, 264]. HRV can be a key indication of time pressure and emotional strain [172]. However, these measures are influenced by the parasympathetic nervous system, which is the portion of the autonomic nervous system that conserves energy such as slowing heart rate when the body is low on energy [155]. PPG measures have an ample body of experimental results in complex scenarios. This is largely due to its association with level of stress [129] and its relative ease of measurement (e.g., with a chest strap or wrist worn device). For instance, Ivarsson et al. investigated how PPG changed in response to violent video games and sleep afterward [125]. Sakuragi et al. investigated how heart rate variability changes in response to watching different genres of videos (e.g., comedy, tragedy, thriller) [213]. McDuff et al. showed that measuring PPG and breathing from a camera could be used to understand measures typically associated with stress, although correlation to actual stress was not shown to be significant [156,157].

5.2. Electro-dermal Activity

Electro-dermal Activity (also known as galvanic skin response) is measured through two electrodes situated on the band of the E4 device. Skin conductance is measured between these two electrodes in microsiemens ($\mu$S). EDA is a known indicator of stress, arousal, and cognitive load [25,163]. Like heart rate, Kahneman showed that tracking of workload changes is possible using EDA as a proxy [131]. Further, it is the only physiological measure that is not contaminated by the parasympathetic nervous system; and is therefore a key indication of the body’s fight or flight response [25,155].
The EDA complex can be broken down into two main components: tonic and phasic. The tonic component is the smooth underlying, slowly changing, and continuous physiological response. The phasic component is the rapidly changing or high frequency portion of the EDA signal. There are several measures derived from these components:

- **Skin conductance level (SCL)** Tonic level (slow response) of electrical conductivity of skin
- **Skin conductance response (SCR)** Phasic change (rapid response) in electrical conductivity
- **Event-related SCR (ER-SCR)** SCRs that can be associated with a specific stimuli or event
- **Non-specific SCR (NS-SCR)** SCRs that occur in the absence of identifiable stimuli (1 to 3 times a minute) [254]

SCL can be measured in two ways: (1) by averaging the tonic signal over a specified period of time (from tens of seconds to tens of minutes), or (2) by taking the frequency of the non-specific SCRs, and then averaging them over the specified time. Averaging the tonic signal is most common because it does not require the attribution of SCLs to certain stimuli. SCL is the absence of a discrete event or stimuli; it varies over time; and it depends on both the environment and psychological state; varying from day-to-day [25].

**ER-SCRs** are defined as the first significant deviation, exceeding a threshold, in the signal one to three seconds from the event-related stimuli, which can include sight, sound, smell, and a cognitive process that can precede event such as anticipation and decision-making [25,35], Figure 5.1. Of note, roughly 10% of the population are non-responders, meaning they do not exhibit any discernible change in EDA as a result of cognitive processes. Serious anxiety can also lead to significant changes in EDA measures on one wrist, but not the other wrist [25,35].
5.3. Peripheral Skin Temperature

Peripheral Skin Temperature is measured via a thermophile located in the housing of the E4 wristband. Temperature is expected to fluctuate when a pilot is flying test points, which could be an indicator of stress. Change in body temperature has been shown to have a direct connection with human perception of duration. That is, increasing body temperatures lead to an increase in apparent duration, and accordingly decreasing temperatures slow it [12, 107]. Hancock et al. suggested this “influence underlies performance variation on a number of tasks including sustained attention” [108]. Further, Hancock et al. suggested tympanic temperature, deep auditory canal temperature (ACT), may be a way to measure cognitive load [109]. Citing Ursin and Ursin’s reasoning [258] they argued the central nervous system under a workload will initiate increased activity. There have been several indirect observations that support this viewpoint [106, 107, 144, 200]. Hancock et al. recommended evaluating the rate of temperature change to “ameliorate the effect of individual difference inherent in all physiological parameters” [109]. To this end, both Haapalainen et al. (heat flux) and Abdelrahman et al. (thermal imaging) have shown body temperature of some form to be a viable modality for measuring cognitive load.
5.4. Wrist Acceleration

Wrist Acceleration is used for assessing movements of the wrist in all three-axes. An objective of the acceleration measurement in our 10-subject feasibility study was to establish the confidence with which other measures are collected. Large, forceful motions degrade the ability of the sensors to capture a number of PPG-based measurements. This was also a contributing factor in the decision to place the E4 on the pilot’s throttle hand. However, wrist acceleration can be an indicator of cognitive load in our application because of the nature of motion for flight maneuvers. That is, the wrist acceleration can be a proxy for a number of pilot actions in the flight scenario, as it may capture nuanced indications of activity or aircraft control. For example, Gray showed that the 'effort' that a pilot puts into manipulating an aircraft can be observed through the concept of inceptor pilot workload [99]. Where an inceptor is just a control input such as the stick. This workload is measured through a combination of two independent variables duty cycle and aggressiveness; and they can be evaluated from the origin of the plot between these two variables, where by duty cycle is the percentage of time that the pilot is manipulating the stick with either force or position. And aggressiveness — using the simple assumption that the faster the pilot is moving the inceptor, the harder he is working” — is the root-mean squared per-second average of the inceptor rate-of-change, position or force [99].
5.5. Gaze Metrics

The gaze metrics included both pupillary response and eye blinks. Eye blink studies have demonstrated there is a correlation among tasks that require attention and fewer eye blinks and/or shorter duration blinks [20,94,249,277]. Chen et al. cited a relationship among cognitive demands and detection, identification [85] and mental arithmetic tasks [253]. Chen et al. found pupillary response and eye blink rate to have low mutual correlation with good cognitive load discrimination within a subject [44]. Further, they classified pupillary response and blink rate between two task levels with an accuracy of 84%, intra-subject [43].

There is a rich history correlating cognitive load with involuntary pupillary responses. Kahneman, in his seminal work in cognitive psychology [131] found pupillary dilation increases proportionally to mental working load in response to various tasks including memorization tasks and visual search tasks. Furthermore, a strong correlation exists between pupillary response and many mental processes, including arousal and interest level (Hess and Polt 1969). Iqbal et al. [15, 123, 124] determined that pupillary dilation is a real-time measure of mental load, showing that pupil dilation can predict the level of attention with which a user interacts with a user interface. She hypothesized that the demand on the users working memory from a user interface results in elevated cognitive load. Bailey and Iqbal [15] have shown that larger tasks should be broken down hierarchically into sub-tasks so that the pupillary dilation peaks are representative of simple tasks. It is important to note the validity of using eye-tracking in lieu of more expensive pupillometers. Klingner (2008) conducted an extensive study to validate the use of a remote eye-tracker for assessing cognitive load, showing that affordable, high-resolution eye-trackers are viable alternatives to pupillometers. Wangwiwatanna et al. verified the ability of eye-trackers to capture cognitive load as well as pupillometers [271].
5.6. Conclusions

Physiological measures beyond pupillary response have shown that pupillary response changes consistently during various activities including word recognition [267], short duration auditory stimuli [87, 185], and in response to pain [74]. However, little work has been done regarding the capturing of pupillary response during more complex, longer duration activities such as during flight simulation. This gap is to be expected as previous studies are geared toward uncovering artifacts of human cognition and informing the theory of cognition. Some work has been conducted. For example, Fukuda et al. studied the effects of pupil dilations and running memory in a long duration task. Their work suggests that assessing the performance of an individual’s ability to “stay-on-task” may be possible through pupillary monitoring [89]. However, their research results were limited by small sample size and relatively few methodologies were tested.

We build upon research for measuring cognitive workload and methods for acquisition of physiological measures during longer duration tasks in order to make them more appropriate for learners during simulated flight. In particular, we investigate methods of combining these modalities using deep learning techniques and convolution, which may more properly fuse the sensor information for use in a cognitive load classifier.
The overall goal of the feasibility studies was to assist in the development of a large-scale experiment leading to the creation of an inference system that can reliably measure cognitive load, gaze patterns, and potentially situational awareness in near real-time using various wearable and environmental biometric sensors. The feasibility studies were used to assess characteristics of the sensing devices and the capture software, and to inform appropriate difficulty levels for the various tasks. Further, through these studies we needed to ensure data can be collected synchronously during both flight and flight simulation with minimal noise sources. To this end, we designed and conducted a study in partnership with L3Harris using a mixed-reality flight simulator. We also participated in a study conducted in actual flight on a military cargo aircraft at Edwards AFB evaluating the E4 wristband. The Objective Measures of Pilot Workload study, an Air Force study, was conducted aboard a C-17 cargo aircraft in actual flight at Edwards Air Force Base [127]. First, we discuss the details of the Empatica E4 sensor suite utilized for both studies, followed by methodologies for the L3Harris Feasibility Study and the Edwards Objective Measures of Pilot Workload Study. Then we look at the results of both studies. Finally, we conclude with a summarization of the feasibility studies and results.

6.1. Empatica E4

In order to keep the form factor for the biometric measurements compact, an all-in-one, wrist-worn device, the Empatica E4 wrist band, was selected. It was developed for research studies aimed at assessing emotional context. The device was issued with FDA approval for monitoring seizures, but the research-based device is simply used as a data collection instrument. A number of sensors are Incorporated on the device, including photoplethysmography
(PPG), electro-dermal activity (EDA), skin surface temperature, and wrist acceleration. The specifics of each modality are discussed in Chapter 5.

It is important to note that the E4 wristband offers two methods for storing and retrieving sensed data. This is done by either storing the data on the device itself for later retrieval, or by streaming the information to another device such as an iPad over Bluetooth. In order to retrieve the data from the E4 after an experiment, Empatica requires the stored data be uploaded to Empatica servers before retrieving it in a cleaned and formatted manner; and given the sensitivity of the data, this is not desired. Further, this method is a post-recording method as data is not available in real-time. For these reasons the streaming method was chosen for this experiment, as it enables immediate access, and we can resolve timing issues based on the received Bluetooth packets.

6.1.1. E4 Timestamp Drift

The Empatica E4 has an issue measuring time. It has tendency to speed time up when running for longer periods. Under 30 minutes the time error is negligible. As time progresses the E4’s clock speeds up, creating material errors. During both flights, the E4 sped up to a few seconds faster than the sensitive instrumentation (SI) on the aircraft and GPS time. Care was taken to ensure that the instrumentation rack was not faulty and that it was reporting accurate time. Further, the time hack taken at the beginning of the flight with the test conductor and the one after showed the same interval. Flight 1 drifted 2.895 seconds faster than SI over a 5.5 hour period. Flight 2 drifted 3.563 seconds faster than SI over a 3.0 hour period.

Additional tests with the E4 on the ground have revealed varying time speedups. This error can occur when using streaming mode. The E4 takes an initial time hack with the streaming device upon connecting. Sample timestamps of the first and any further packets are calculated from the reference timestamp and the sample frequency of the respective stream. The longer the E4 is used during the same session, the further the drift. If there is a disconnect and a subsequent reconnection is made, then a new time hack is taken upon
connection and the process starts over. Empatica confirmed this issue in a statement:

*The time drift exists because we need to use an estimated sampling rate for the time-stamping of the samples. (See [http://developer.empatica.com/windows-streaming-server-data.html](http://developer.empatica.com/windows-streaming-server-data.html)). Due to various factors (e.g. ambient temperature) the estimated sampling rate can differ from the actual sampling rate, introducing a small drift. If you need higher accuracy over a longer time streaming/recording, you may employ additional methods to improve the timestamping such as adding tags (button press) to help synchronise timestamp.*

During data reduction, care was taken to adjust clocks to lineup with other datasets such as the aircraft data as the time delta increased over time.

### 6.2. L3Harris Feasibility Study Methodology

The L3Harris feasibility study was a repeated measures human subjects experiment incorporating a primary flight task and cognitive load-inducing, secondary task. The Empatica E4 wristband was worn by all participants on their throttle hand. Scenarios were flown on L3Harris’s Pelican 2 simulator with an FX3 gaze tracker [1]. Two tasks were conducted for any given scenario: (1) a primary flying task, and (2) a secondary workload inducing task. At the start of each experiment, a calibration protocol was carried out with the aim of assessing the participant’s baseline ability to complete the secondary cognitive load-inducing task; otherwise, all scenarios include either the primary task itself, or the primary and secondary task combined. The primary task was defined by L3Harris and consists of five successive scenarios for use on L3Harris’s Pelican 2 simulator. The Pelican 2 used for this experiment was not easily reset for multiple aircraft positions for the chosen five scenarios. Therefore, each successive scenario needed to be completed in order, given that each scenario’s end sets up the aircraft position for the next scenario.
The five scenarios include:

<table>
<thead>
<tr>
<th>Event</th>
<th>Name</th>
<th>Parameters</th>
<th>Stop Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Final approach and landing</td>
<td>Land at 140 KIAS and decelerate to 100 KIAS</td>
<td>1:30</td>
</tr>
<tr>
<td>2</td>
<td>Climb out to 3000'</td>
<td>5 DEG climb out, accelerate to 450 (full throttle), level off at 3000'</td>
<td>2:30</td>
</tr>
<tr>
<td>3</td>
<td>Immelmann + high speed low pass</td>
<td>Perform Immelmann, full throttle low pass on full length of runway at 50' (radar altitude)</td>
<td>3:40</td>
</tr>
<tr>
<td>4</td>
<td>Fly vertical + level off</td>
<td>90 DEG vertical climb, level off when speed is 200 KIAS, then accelerate to 300 KIAS</td>
<td>4:20</td>
</tr>
<tr>
<td>5</td>
<td>Inverted dive and high-speed landing</td>
<td>Invert plane, dive back to Randolph, no throttle, drop gears for drag, land</td>
<td>5:30</td>
</tr>
</tbody>
</table>

Table 6.1: L3Harris Feasibility Study Maneuvers

A secondary task was used to artificially increase cognitive load in participants. The chosen secondary task induces cognitive load by presenting the pilot with aural stimuli requiring a manual response, a tap, on a touch screen. This task is commonly referred to as an “oddball task” [72, 244]. The participant was presented a tone at a rate of every two seconds. At semi-random intervals a higher frequency tone was played, the oddball tone. When the oddball was played, the participant tapped the touchscreen, which is on or near the knee closest to the throttle. When a regular tone was played, the participant was instructed to do nothing and continue with the primary task. For this experiment the regular tone frequency was 600 Hz and the odd ball tone frequency was 800 Hz. This task is known to invoke cognitive load and requires a moderate level of concentration to complete. By using this secondary task, cognitive load and potentially inattentional deafness [37] can be induced, through attentive visual stimulus (primary task) and aural stimuli (secondary task). By adjusting the frequency at which the secondary task is administered, the cognitive load can be modified.

Participants started the experiment by entering their demographic information. The participants repeated the five scenarios twice, once with secondary tasks and once without. The order was chosen at random in order to prevent ordering effects; however, given constraints on the simulator, the five scenarios were required to be completed in flight order. After each run, the participants reported cognitive loads using two instruments: the NASA TLX [113] and Bedford Pilot Workload Scale [208]. NASA TLX consists of several questions, on a 0-100 scale, that assess mental demand, physical demand, temporal demand, effort exerted,
the pilot’s estimation of performance in accomplishing the task, and the frustration experienced. The Bedford workload scale is a decision tree of questions that results in a workload rating between 1-10. A pilot answers a series of questions to eventually come to a concluding Bedford workload rating. For more information on subjective scales see Section 4.2.3. These subjective scales were utilized to verify that the secondary induced cognitive load was effective, and to verify that the induced level of cognitive load was reasonable.

The application discussed in Appendix D serves both as the interface for the secondary task as well as the acquisition of the sensor data from the E4 wristband sensor suite. Gaze data was reported by the gaze tracker in a separate CSV file. The feasibility of Tobii eye trackers have already been validated in previous SMU research [196,197,270,271]. Therefore, the focuses of these studies were on the Empatica E4. The IRB application and relevant protocols were submitted and approved on September 21, 2018, application H18-105-LARE.

6.3. Edwards Objective Measures of Pilot Workload Study Methodology

The Edwards Objective Measures of Pilot Workload study was a feasibility study to demonstrate the utility of physiological sensor systems as a flight test data source for assessing pilot workload. The SMU AT&T Center for Virtualization was one of three institutions that participated the study. With our Empatica E4 capture software, we participated in two test flights. A key goal was to have the test pilot flying wear as many sensors as possible given the limited number of sensors and constraints of flight. Ultimately, this feasibility study was a data capture experiment evaluating the capacity to collect such data in real-flight; but also it was used for assessing the practicality and wear-ability of the sensors. The study was conducted in real-flight on a military cargo aircraft at Edwards AFB, where we evaluated the Empatica E4 wristband.

Multiple maneuvers were flown in order to collect data and potentially assess pilot workload. Aerial refueling and offset landing tests were expected to induce high workload. Normal flight profile maneuvers such as climb, cruise, and decent were expected to induce low or operational workload. The test conditions were operationally representative of real flight.
Data collection was conducted on an instrumented C-17A, a large cargo aircraft. The IRB application and relevant protocols were submitted and approved through the Air Force Research Laboratory (AFRL) on June 28, 2018. The IRB at Southern Methodist University approved through reciprocity. The ARFL IR ID protocol number is FWR20180152N. Flight test was conducted by the 418th Flight Test Squadron in accordance with test plan, 412TW-TP-18-47, Objective Measures of Workload Feasibility Study [127].

While climb, cruise, and decent were collected along the normal flight profile, three maneuvers were flown multiple times in order to collect as much high workload data as possible. These maneuvers included [127]: (1) aerial refueling boom limits demonstration, (2) aerial refueling stationkeeping, and (3) lateral offset landing test maneuver. The potential for baseline physiological data was also possible during low workload tasks such as periods of rest between test points. The three maneuvers flown were as follows, note the C-17A test aircraft was the receiving aircraft in the first two maneuvers:

1. **Aerial Refueling Boom Limits Demonstration:** The goal was to fly this maneuver for a minimum of five minutes. Once cleared by the tanker to precontact and contact positions, the boom operator of the tanker made contact with the receiving aircraft and directed the receiving aircraft through a boom limits demonstration; this included all three boom axes — azimuth, elevation, and extension. The test aircraft held boom position within three degrees or two feet of each limit, until stabilized. The pilot flying held this position for five seconds. Figure 6.1 depicts the six boom positions for the boom demo.

2. **Aerial Refueling Stationkeeping:** As with the Boom Limits Demo, the stationkeeping maneuver began with clearance from the tanker to precontact and contact positions. The boom operator will then make contact. For a minimum of five minutes, the receiving aircraft remained in contact, also known as stationkeeping. The receiving aircraft, remaining in contact with the tanker, was stabilized in straight and level flight. The tanker and the receiving aircraft banked 30°. This was held through at least 90° of heading change. Both aircraft returned to straight and level flight. At which point the maneuver was completed had five minutes or more elapsed.
3. **Lateral Offset Landing Test Maneuver:** The pilot flying conducted an approach to a runway with a 300ft lateral offset distance. A touchdown point on the runway was identified at the pilot flying’s discretion. At 500 ft above ground level (AGL), the pilot flying initiated recovery to runway centerline. They then lined up and stabilize on their desired touchdown point by 300 ft AGL. ‘Go Around’ was called if they were not stabilized before 300 ft AGL. If the pilot flying was stabilized before to 300 ft AGL, the maneuver could be terminated with either a touch and go or a full-stop landing.

![Boom Perspective](image)

Figure 6.1: Boom Limits Demonstration — Points five and six are inner and outer limits [127]

### 6.4. Analysis

**Accelerometer Analysis**

Throttle hand acceleration in all 3-axes can be analyzed individually or as a magnitude. Here, we used total magnitude, where magnitude = \( \sqrt{x^2 + y^2 + z^2} \). Both the range of magnitudes and number of occurrences with respect to throttle movement are key measures.
Stress can be evaluated by examining the PPG, specifically the heart rate variability (HRV). A necessary component of HRV is inter-beat interval (IBI). IBI is the measured interval in time between heart beats. The Empatica E4 wristband uses a built-in proprietary algorithm on the PPG signal for determining the inter-beat interval. The algorithm strives for extremely accurate non-interpolated data; but it achieves this at the cost of being susceptible to large swings in acceleration. With heavy movement the algorithm cannot determine IBI because the PPG signal becomes too noisy. Therefore, during periods of heavy motion, the E4 did not report IBI. The E4 employs a red-light beam in the pulse oximeter; this enables the E4 to reduce some motion noise from the PPG signal [76, 77]. The E4 wristband does not interpolate IBI; and it can handle small motion artifacts, which is important in ensuring accurate data.

The simplest way to measure HRV is to take the standard deviation of IBI derived from the PPG signal. This is known as SDNN. Where SD stands for standard deviation and NN stands for normal-to-normal intervals, or more simply, inter-beat intervals. “NN intervals [are] interbeat intervals from which artifacts have been removed, [and] RR intervals [are] interbeat intervals between all successive heartbeats” [231]. With IBI from the PPG data, the standard deviation can be calculated on IBI data over a period of time. Standard practice is at least five minutes. Because the “variance is mathematically equal to the total power of spectral analysis, SDNN [or SDRR] estimates the cyclic components responsible for vari-
ability in the period of recording” [254]. As the period of time decreases, both cycle length estimation and total variation decrease. Accurate HRV can only be measured when comparing observations of the same length. With applied stress, the sympathetic nervous system responds by stabilizing the heart rate and a decrease in heart rate variance occurs [254]. Therefore, lower heart rate variance is a proxy for stress and fatigue [117,172].

Electro-dermal Activity Analysis

As discussed in Section 5.2, the EDA complex can be broken down into two main components: tonic and phasic. The tonic component is the smooth underlying, slowly changing, and continuous physiological response. The phasic component is the rapidly changing or high frequency portion of the EDA signal. There are several measures derived from these components including SCL, SCR, ER-SCR, and NS-SCR.

Noisy points of the EDA signal may be removed whenever the magnitude of the accelerometer data is too high or whenever the temperature is less than body temperature. Tonic and phasic are separated from the original EDA waveform by using a finite impulse response (FIR) filter, both forward and backward, yielding the phasic waveform — this is known as zero phase filtering [103]. Tonic is then calculated by subtracting the phasic waveform from the original EDA waveform. The filter coefficients employed are calculated from a Butterworth filter [34] with two Hz cutoff. Butterworth filter design is achieved using a least squares estimate of the pass and stop bands. In our analysis, this achieved minimal pass band ripple effects with sufficient damping. The coefficients were applied forward and backward to intensify the effect as well as remove any delay from the filtering process. Once processed and separated, metrics can be taken. The calculated filter coefficients are included here for reproducibility:

<table>
<thead>
<tr>
<th>Filter Coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001398398042</td>
</tr>
<tr>
<td>-0.002980425267</td>
</tr>
<tr>
<td>-0.038101343570</td>
</tr>
<tr>
<td>-0.109709686652</td>
</tr>
<tr>
<td>0.849609375000</td>
</tr>
<tr>
<td>-0.109709686652</td>
</tr>
<tr>
<td>-0.038101343570</td>
</tr>
<tr>
<td>-0.002980425267</td>
</tr>
<tr>
<td>0.001398398042</td>
</tr>
</tbody>
</table>
Secondary Task Analysis

The oddball task was utilized during the L3Harris feasibility study only. The oddball task’s primary parameters included the number of misses, response times, and correctness. A miss was defined as receiving no response given an oddball tone, and then a consecutive tone was played. Response time can be measured by calculating the difference from the time the screen is tapped to the time the oddball tone was played. Correctness was defined as a hit, miss, or multi-tap. A hit consisted of tapping the screen before the next consecutive tone in response to an oddball tone. Where a miss is the opposite, not tapping. A multi-tap happens when the screen is tapped more than once for the same stimulus. With a basic statistical analysis of these parameters and the participant’s self-scored NASA TLX and the Bedford workload scales, a better picture of the workload induced during a scenario could be understood.

6.5. L3Harris Feasibility Study Results

The L3Harris feasibility study was conducted in October of 2018 at L3Harris Link, Arlington. Data were collected for ten participants: six pilots and four novices.

Acceleration

In analyzing acceleration, we used a morphological peak finding method, whereby signal dilation measures were used to find central peaks in a moving window. Considering the data from all the participants, both pilot and novice, extreme magnitudes are rare, but they do occur. The average count of magnitude peaks greater than 1.3 G and 1.4 G for all subjects and scenarios, both with and without the secondary task were 6.5 and 4.3 peaks, respectively. If the secondary task data are removed, then the averages are reduced to 4.5 and 1.8 G peaks. When peaks did exceed 1.5 G, the peak magnitude trended towards 2.0 g as some participants did have trouble. Participant eight struggled on the final approach and landing with the oddball task, as explained in more detail below.
Table 6.3: L3Harris Feasibility Study Magnitude Statistics

<table>
<thead>
<tr>
<th>Max Magnitude</th>
<th>&gt;1.3G</th>
<th>&gt;1.4G</th>
<th>&gt;1.5G</th>
<th>Window Size</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.013</td>
<td>16</td>
<td>12</td>
<td>11</td>
<td>30.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2.013</td>
<td>13</td>
<td>11</td>
<td>10</td>
<td>30.0</td>
<td>10.0</td>
</tr>
<tr>
<td>2.013</td>
<td>12</td>
<td>8</td>
<td>7</td>
<td>30.0</td>
<td>20.0</td>
</tr>
<tr>
<td>1.887</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>30.0</td>
<td>30.0</td>
</tr>
<tr>
<td>1.748</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>30.0</td>
<td>50.0</td>
</tr>
<tr>
<td>1.987</td>
<td>9</td>
<td>6</td>
<td>6</td>
<td>28.0</td>
<td>60.0</td>
</tr>
<tr>
<td>1.987</td>
<td>8</td>
<td>7</td>
<td>5</td>
<td>18.0</td>
<td>70.0</td>
</tr>
<tr>
<td>1.785</td>
<td>5</td>
<td>5</td>
<td>3</td>
<td>8.0</td>
<td>80.0</td>
</tr>
</tbody>
</table>

(a) Participant eight: final approach and landing with secondary task

<table>
<thead>
<tr>
<th>Max Magnitude</th>
<th>&gt;1.3G</th>
<th>&gt;1.4G</th>
<th>&gt;1.5G</th>
<th>Window Size</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.417</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>30.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1.263</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30.0</td>
<td>10.0</td>
</tr>
<tr>
<td>1.351</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30.0</td>
<td>20.0</td>
</tr>
<tr>
<td>1.209</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30.0</td>
<td>30.0</td>
</tr>
<tr>
<td>1.351</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30.0</td>
<td>40.0</td>
</tr>
<tr>
<td>1.290</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30.0</td>
<td>50.0</td>
</tr>
<tr>
<td>1.351</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30.0</td>
<td>60.0</td>
</tr>
<tr>
<td>1.351</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>30.0</td>
<td>70.0</td>
</tr>
<tr>
<td>1.202</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30.0</td>
<td>80.0</td>
</tr>
<tr>
<td>1.202</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30.0</td>
<td>90.0</td>
</tr>
<tr>
<td>1.202</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30.0</td>
<td>100.0</td>
</tr>
</tbody>
</table>

(b) Participant six: final approach and landing

Table 6.3 (a) presents magnitude statistics for participant eight’s 80-second run, while on the scenario final approach and landing. Each row is calculated on a 30-second sliding window. Each consecutive row is separated by 10 seconds and overlaps 20 seconds of the previous row. This is an example of poor fine motor control with the throttle. However, most participant data were in alignment with participant 6, a pilot Table 6.3 (b). During participant six’s 100-second run for final approach and landing most peak magnitudes were under 1.4 Gs.

Figure 6.3: L3Harris Feasibility Study PPG, IBI, and Acceleration: Magnitude baseline, participant six
Photoplethysmography and Heart Rate Variability

Unfortunately, direct measurement of HRV was not possible because the pulse oximeter is too susceptible to motion, and Empatica’s proprietary algorithm does not compensate enough for moderate-to-heavy motion. A key factor is that the algorithm does not interpolate. Often a 30-second period would yield only a handful of inter-beat-interval data points. Further, those handful of data points could be disjointed and were not always consecutive. To obtain true HRV a continuous recording period was required, and the variation must only be compared with other HRV measures of the same recording length. However, an approximation of HRV may be good enough to still take advantage of the sympathetic nervous system’s response to stress. The majority of the L3Harris feasibility study data is similar to what is depicted in Figure 6.3; while noisy and yielding very few E4 IBI data points, the raw PPG is very clearly still usable.

Table 6.4 describes the IBI statistics for the baseline scenario of participant six over the two-minute period of recording. Each row represents 30-seconds, with each consecutive row calculated 10 seconds later, overlapping the previous two rows. There is a slight decrease in the standard deviation as the baseline scenario progresses.

Table 6.4: L3Harris Feasibility Study IBI Statistics: Baseline secondary task, participant six

<table>
<thead>
<tr>
<th>IBI Max</th>
<th>IBI Std. Dev.</th>
<th>IBI Count</th>
<th>Window Size</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.906</td>
<td>0.021</td>
<td>14</td>
<td>30.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.953</td>
<td>0.034</td>
<td>7</td>
<td>30.0</td>
<td>10.0</td>
</tr>
<tr>
<td>0.953</td>
<td>0.019</td>
<td>3</td>
<td>30.0</td>
<td>20.0</td>
</tr>
<tr>
<td>0.953</td>
<td>0.017</td>
<td>9</td>
<td>30.0</td>
<td>30.0</td>
</tr>
<tr>
<td>0.953</td>
<td>0.023</td>
<td>12</td>
<td>30.0</td>
<td>40.0</td>
</tr>
<tr>
<td>0.953</td>
<td>0.023</td>
<td>14</td>
<td>30.0</td>
<td>50.0</td>
</tr>
<tr>
<td>0.922</td>
<td>0.018</td>
<td>8</td>
<td>30.0</td>
<td>60.0</td>
</tr>
<tr>
<td>0.922</td>
<td>0.016</td>
<td>6</td>
<td>30.0</td>
<td>70.0</td>
</tr>
<tr>
<td>0.922</td>
<td>0.017</td>
<td>4</td>
<td>30.0</td>
<td>80.0</td>
</tr>
<tr>
<td>0.922</td>
<td>0.017</td>
<td>4</td>
<td>30.0</td>
<td>90.0</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>0</td>
<td>20.0</td>
<td>100.0</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>0</td>
<td>10.0</td>
<td>110.0</td>
</tr>
</tbody>
</table>
**Electro-dermal Activity**

For the feasibility study both the tonic and phasic responses were observed, specifically SCL and ER-SCR. Usable SCL is potentially too slow, given it can manifest a response within 10s of seconds to 10s of minutes. However, in our tests it was found that the SCR measure was more reliable as they manifest within one to three seconds of a stimulus event. SCRs occurred for all participants, both pilots and novices. In general SCRs occurred less frequently in pilots than novices. However, it was observed that in one case where an experienced pilot seemed relatively flat with respect to SCL, there was a large level of SCR activity. In this case participant Two on the final approach and landing scenario, Figure 6.4 and Table 6.5 show this unique case.

![Figure 6.4: L3Harris Feasibility Study EDA: Final approach and landing, participant two](image)

**Table 6.5: L3Harris Feasibility Study EDA Statistics: Final approach and landing, participant two**

<table>
<thead>
<tr>
<th>EDA SCR Max</th>
<th>SCR Count</th>
<th>EDA SCL</th>
<th>Tonic Slope</th>
<th>Window Size</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.567</td>
<td>9</td>
<td>8.237</td>
<td>7.403</td>
<td>30.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.567</td>
<td>9</td>
<td>8.303</td>
<td>-0.157</td>
<td>30.0</td>
<td>10.0</td>
</tr>
<tr>
<td>0.544</td>
<td>7</td>
<td>8.342</td>
<td>-9.380</td>
<td>24.0</td>
<td>20.0</td>
</tr>
<tr>
<td>0.544</td>
<td>4</td>
<td>8.198</td>
<td>-19.906</td>
<td>14.0</td>
<td>30.0</td>
</tr>
<tr>
<td>0.364</td>
<td>1</td>
<td>7.996</td>
<td>91.209</td>
<td>4.0</td>
<td>40.0</td>
</tr>
</tbody>
</table>
Given the nature of this experiment there was no direct way to isolate NS-SCR and were considered noise in the ER-SCR data. However, one can correlate an event-driven stimulus such as landing an airplane. You can rule out NS-SCR, such as in the case of participant three, Figure 6.5 and Table 6.6, an experienced pilot, on the final approach and landing scenario where he had an event-related response upon landing, 0.058 µS.

Because EDA is a known indicator of stress, arousal, and cognitive load [25,35,163], and more specifically ER-SCRs can be a physiological response to cognitive processes [25]. Table 6.7 reveals the maximum SCR each participant in the pilot study triggered. The maximum response observed across all pilots and maneuvers was .619 µS. It is clear that intensity is related to the strength of the bodies’ fight or flight response.

![Figure 6.5](image)

**Figure 6.5:** L3Harris Feasibility Study EDA: Final approach and landing, participant three

<table>
<thead>
<tr>
<th>EDA SCR Max</th>
<th>SCR Count</th>
<th>EDA SCL</th>
<th>Tonic Slope</th>
<th>Window Size</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>0</td>
<td>0.604</td>
<td>0.254</td>
<td>30.0</td>
<td>0.0</td>
</tr>
<tr>
<td>0.058</td>
<td>1</td>
<td>0.609</td>
<td>-0.048</td>
<td>30.0</td>
<td>10.0</td>
</tr>
<tr>
<td>0.058</td>
<td>1</td>
<td>0.609</td>
<td>-0.071</td>
<td>21.0</td>
<td>20.0</td>
</tr>
<tr>
<td>0.058</td>
<td>1</td>
<td>0.607</td>
<td>0.571</td>
<td>11.0</td>
<td>30.0</td>
</tr>
<tr>
<td>None</td>
<td>0</td>
<td>0.622</td>
<td>1.843</td>
<td>1.0</td>
<td>40.0</td>
</tr>
</tbody>
</table>

**Table 6.6:** L3Harris Feasibility Study EDA statistics: Final approach and landing, participant three
Table 6.7: L3Harris Feasibility Study Peak SCRs — All Subjects

<table>
<thead>
<tr>
<th>Participant</th>
<th>Max SCR (µS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.211</td>
</tr>
<tr>
<td>2</td>
<td>0.19</td>
</tr>
<tr>
<td>3</td>
<td>0.104</td>
</tr>
<tr>
<td>4</td>
<td>0.16</td>
</tr>
<tr>
<td>5</td>
<td>0.3</td>
</tr>
<tr>
<td>6</td>
<td>0.586</td>
</tr>
<tr>
<td>7</td>
<td>0.619</td>
</tr>
<tr>
<td>8</td>
<td>0.185</td>
</tr>
<tr>
<td>9</td>
<td>0.101</td>
</tr>
<tr>
<td>10</td>
<td>0.5</td>
</tr>
<tr>
<td>Average</td>
<td>0.4956</td>
</tr>
</tbody>
</table>

*Secondary Task*

The oddball task’s primary parameters included the number of misses, response time, and correctness. The response times were very consistent from participant to participant. Participant eight had many misses, 39%; but his response time was on par with the rest of the sample population. There appeared to be a slightly negative slope as the participants progressed through the baseline scenario. Otherwise, the response time parameter was largely flat averaging .89 seconds for all participants. Overall, both pilots and novices had no issues completing the oddball tasks while flying the primary task, which was not useful for inducing cognitive load. Part of this could be driven by the fact that the scenarios were not difficult enough for the experienced pilots. Even so, the novices, except for participant eight, had no problems with the secondary task while attempting to accomplish the hard flying tasks they had never done before. Even worse, there was the potential to induce noise in the data, given how EDA responded to oddball stimuli. We found that the phasic response of EDA synchronized with the aural stimulus, thus inducing an artificial trigger during the experiment. This phasic peak would not be present without the oddball task, and therefore we anticipated that the results of performing further analysis on the secondary task data would not lead to generalizing results.
While utilizing the oddball task, there were clear SCRs correlating with the playing of a tone. As was the case for participant three during the baseline scenario, Figure 6.6. While this was clearly a cognitive response, this has a potential of inducing noise when training a deep-learning model for the primary task’s workload. The oddball task did not offer new information into the workload of the participant during a scenario; therefore, it is likely better to focus efforts on defining scenarios that generate maximum workload, free of secondary task interference, and building a deep learning model that can infer the workload in-between baseline and maximum workload.

Figure 6.6: L3Harris Feasibility Study ER-SCRs: Oddball baseline scenario correlated to oddball tones, participant three

6.6. Edwards Objective Measures of Pilot Workload Results

Flight test was conducted on August 28 and on 30 of 2018 at Edwards Air Force Base, CA. Data were collected on five experienced pilots, covering 31 test points and five pilots (A-E).
As with the L3Harris feasibility study, Section 6.5, a peak counting method was used, where signal dilation measures were used to find central peaks in a moving window. Considering the data from all five pilots, extreme magnitudes were again rare. However, they can occur. The low magnitude vibration, < 1.2 G, was common and likely attributed to the aircraft in motion and yoke oscillation during refueling, see Figure 6.7. The average count of magnitude peaks for a 30-second period over all test points were: 3.65, 1.2, and 0.043 for peaks greater than 1.3, 1.4, and 1.5 g, respectively.

### Table 6.8: Edwards Objective Measures Study Magnitude Statistics

#### (a) Pilot A: AR station keeping, flight 1

<table>
<thead>
<tr>
<th>Max Magnitude</th>
<th>Mean Magnitude</th>
<th>Standard Deviation</th>
<th>&lt;1.2 G</th>
<th>&lt;1.3 G</th>
<th>&lt;1.4 G</th>
<th>&lt;1.5 G</th>
<th>Window Size</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.315</td>
<td>0.917</td>
<td>0.261</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>22.3</td>
</tr>
<tr>
<td>1.432</td>
<td>0.959</td>
<td>0.284</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.523</td>
<td>0.991</td>
<td>0.036</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.632</td>
<td>0.973</td>
<td>0.073</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.438</td>
<td>0.971</td>
<td>0.089</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.524</td>
<td>0.930</td>
<td>0.277</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.632</td>
<td>1.008</td>
<td>0.001</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.418</td>
<td>0.995</td>
<td>0.071</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.519</td>
<td>1.014</td>
<td>0.074</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.628</td>
<td>1.003</td>
<td>0.087</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.434</td>
<td>0.989</td>
<td>0.068</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.524</td>
<td>1.023</td>
<td>0.072</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.633</td>
<td>1.006</td>
<td>0.078</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.438</td>
<td>1.013</td>
<td>0.089</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.524</td>
<td>1.022</td>
<td>0.073</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.633</td>
<td>1.014</td>
<td>0.072</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.438</td>
<td>0.989</td>
<td>0.068</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.524</td>
<td>1.023</td>
<td>0.072</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.633</td>
<td>1.014</td>
<td>0.072</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.438</td>
<td>0.989</td>
<td>0.068</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.524</td>
<td>1.023</td>
<td>0.072</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.633</td>
<td>1.014</td>
<td>0.072</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.438</td>
<td>0.989</td>
<td>0.068</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.524</td>
<td>1.023</td>
<td>0.072</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
<tr>
<td>1.633</td>
<td>1.014</td>
<td>0.072</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>30</td>
<td>30.0</td>
</tr>
</tbody>
</table>

#### (b) Pilot E: AR station keeping, flight 2

<table>
<thead>
<tr>
<th>Max Magnitude</th>
<th>Mean Magnitude</th>
<th>Standard Deviation</th>
<th>&lt;1.2 G</th>
<th>&lt;1.3 G</th>
<th>&lt;1.4 G</th>
<th>&lt;1.5 G</th>
<th>Window Size</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.237</td>
<td>0.994</td>
<td>0.089</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>22.3</td>
</tr>
<tr>
<td>1.248</td>
<td>0.983</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
<tr>
<td>1.365</td>
<td>0.994</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td>33.0</td>
</tr>
</tbody>
</table>

Figure 6.7: Edwards Objective Measures Study Examples of acceleration magnitude: (Top) Pilot A, Flight 1, AR boom limits (Bottom) Pilot D, Flight 2, AR boom limits
Figure 6.8: Edwards Objective Measures Study PPG, IBI, and Magnitude

(a) Pilot A: Normal approach landing (PPG, IBI, and Magnitude)

(b) Pilot E: Climb (PPG, IBI, and Magnitude)

(c) Pilot E: Cruise (PPG, IBI, and Magnitude)
Table 6.8 (a) presents magnitude statistics for pilot A’s test point at 22:48:50Z, the aerial refueling (AR) station keeping test maneuver, until disconnect at 22:56:30. Table 6.8 (b) presents statistics for pilot E’s test point at 18:04:02Z, also AR station keeping, until disconnect at 18:10:52Z. Each row was calculated on a 30-second sliding window. Each consecutive row was separated by 15 seconds and overlaps 15 seconds of the previous row. Figure 6.8 (a) has very few accelerations that are greater than 1.3g. Whereas the Figure 6.8 (b) has several accelerations counts greater than 1.4g, during the turn. The majority of the remaining test points are similar to the figure on the right and have few magnitudes larger than 1.5.

**Photoplethysmography and Heart Rate Variability**

Similar to the L3Harris feasibility study results, Section 6.5, the pulse oximeter was again too susceptible to motion for the same reasons discussed in the previous section. Given the level of noise outside the lab setting, an approximation of HRV is likely good enough to still take advantage of the sympathetic nervous systems’ response to stress. Table 6.9 represents the majority of the Edwards data. While noisy and with few E4 IBI data points, again the raw PPG is still very usable.

<table>
<thead>
<tr>
<th>Max IBI</th>
<th>Mean IBI</th>
<th>Standard Deviation</th>
<th>IBI Count</th>
<th>Window Size</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0156</td>
<td>0.915</td>
<td>0.078</td>
<td>7</td>
<td>30.0</td>
<td>30.0</td>
</tr>
<tr>
<td>1.125</td>
<td>0.93</td>
<td>0.088</td>
<td>11</td>
<td>30.0</td>
<td>45.0</td>
</tr>
<tr>
<td>1.125</td>
<td>0.938</td>
<td>0.071</td>
<td>11</td>
<td>30.0</td>
<td>60.0</td>
</tr>
<tr>
<td>0.969</td>
<td>0.914</td>
<td>0.033</td>
<td>12</td>
<td>30.0</td>
<td>75.0</td>
</tr>
<tr>
<td>1.031</td>
<td>0.899</td>
<td>0.052</td>
<td>17</td>
<td>30.0</td>
<td>90.0</td>
</tr>
<tr>
<td>1.031</td>
<td>0.9</td>
<td>0.068</td>
<td>20</td>
<td>30.0</td>
<td>105.0</td>
</tr>
<tr>
<td>1.0</td>
<td>0.922</td>
<td>0.06</td>
<td>16</td>
<td>30.0</td>
<td>120.0</td>
</tr>
<tr>
<td>0.984</td>
<td>0.902</td>
<td>0.0466</td>
<td>18</td>
<td>30.0</td>
<td>135.0</td>
</tr>
<tr>
<td>0.984</td>
<td>0.889</td>
<td>0.049</td>
<td>20</td>
<td>30.0</td>
<td>150.0</td>
</tr>
<tr>
<td>0.984</td>
<td>0.905</td>
<td>0.059</td>
<td>0</td>
<td>30.0</td>
<td>158.617</td>
</tr>
</tbody>
</table>
Table, 6.9, describes IBI statistics for the climb test point, pilot E, over a 158 second period of recording. Each row represents 30 seconds, with each consecutive row calculated 15 seconds later, overlapping the previous row. As with the L3Harris study a slight decrease in the standard deviation as the test point progresses is apparent. This test point was chosen because of the small IBI count similarity compared with other test points with large gaps in data.

Electro-dermal Activity

As with the L3Harris feasibility study results are similar. For EDA both the tonic and phasic responses were observed and tonic/skin conductance level was extremely slow. Figure 6.9 shows the tonic portion of the EDA complex slowly rising over 11 minutes during an AR boom limits demo. The SCR measure was a more readily-usable measure, as it manifested within one-to-three seconds of a stimulus event; still both should be utilized for development of a machine learning model. SCRs occurred for all pilots; however, far fewer observations occurred for pilot D from the second flight. Figures 6.10 (a and b) reflect the tonic and phasic responses during AR boom limits demo and AR station keeping test points for Pilot D. While fairly quiet for the majority of both test points, SCRs above .03 were observed during key moments of workload. For the AR boom limits demo, they occurred twice in two groups. The first happened when the pilot was in weather working to regain contact with the tanker for roughly two minutes, ultimately leading to contact at 16:45:53. The second group occurred while pilot D was working his way to position 6, 16:48:58, and eventually disconnecting at 16:49:44 on his return to position 0; peak .242 µS SCR at 17:01.
As with the L3Harris study we were able to rule out NS-SCRs, by correlating an event related stimulus with an SCR. When Pilot A on the 1st flight rotated the aircraft during takeoff, pilot A manifested an ER-SCR of 0.099 $\mu S$, Figure 6.11.
The intensity of the EDA SCR response is directly related to the strength of the body’s fight or flight response. Table 6.10 reveals the maximum SCR each participant in the pilot study triggered. The maximum response observed was .243 $\mu S$.

Table 6.10: Edwards Objective Measures Study Peak SCRs — All Subjects

<table>
<thead>
<tr>
<th>Participant</th>
<th>Max SCR ($\mu S$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.138</td>
</tr>
<tr>
<td>B</td>
<td>0.168</td>
</tr>
<tr>
<td>C</td>
<td>0.24</td>
</tr>
<tr>
<td>D</td>
<td>0.243</td>
</tr>
<tr>
<td>E</td>
<td>0.11</td>
</tr>
<tr>
<td>Average</td>
<td>0.136</td>
</tr>
</tbody>
</table>

6.7. Conclusions

The accelerometer, while adept in capturing wrist motion, is also useful in the task of noise reduction, especially for motion in the PPG signal. It is very clear from the Edwards Objective Measures of Pilot Workload Study that aircraft motion had an effect on acceleration, however subtle. Skin conductance level and the magnitude and frequency of skin conductance responses correlate well with pilot workload over a range of test maneuvers. SCL was slow to respond and might require a memory capable algorithm to be useful in real-time, we will explore deep learning methods. Further, the specificity of these measures over time made it difficult to pinpoint the exact stimulus that resulted in increased workload. Using skin conductance alone may only reveal that certain intervals influenced the overall pilot workload. It is known that with repetition skin conductance responses diminish over time and eventually fade [35]. EDA’s SCL and SCR are only two pieces of the puzzle. We learned that gaze patterns and pupil response could augment skin conductance measures. The ensemble could provide a higher fidelity and near real-time picture of workload. The suite of sensors on the E4, while not perfect, were found to be beneficial.
The feasibility studies were used to both assess characteristics of the sensing devices, the capture software, and inform appropriate difficulty level for the various tasks. Further, through these studies we ensured data can be collected synchronously with both flight and flight simulation with minimal noise sources. While it is clear the Empatica E4 wristband has some utility for assessing pilot workload, further investigation is needed to produce a clearer answer. Most importantly a much larger dataset is needed. We hypothesize a statistically-significant, deep learning model can be built with this type of data. However, more data is necessary, and that data needs a balanced ratio of loaded vs unloaded labels; this would be required to build a successful model. Our full study is discussed in the next chapter.
This chapter covers the full physiological study, that is the methodology utilized in the collection of physiological data for 40 test subjects. The overall objective was to collect a large dataset made up of subject physiological, simulator, and situation awareness global assessment technique (SAGAT) data. The dataset collected was utilized towards the development of a real-time biometric-based human performance inference system. The feasibility studies established the characteristics of the sensing devices, and the capture software. This section discusses the procedures and equipment used to build the larger data set needed for deep learning. First, the sensors and equipment used to capture data are discussed, followed by gaze data format. A description of the methodology is discussed. Finally, the demographics of the study are explained. The data collected in this study are utilized throughout this research to build the classifiers needed that can reliably measure cognitive load, gaze scan patterns, and using wearable and environmental, physiological/biometric sensors.

### 7.1. Sensors and Equipment

For the test/experiment we used a prototype simulator that was provided by *L3Harris Technologies, Link Training & Simulation* — the Blue Boxer Extended Reality Simulator (BBXR) [110]. The Blue Boxer is a portable, mixed reality system that uses virtual reality (VR) and high-precision hand tracking to simulate aircraft flight characteristics for training. Designed to be compact and portable, the system amalgamates physical and virtual mission equipment to simulate the flight environment. In addition to the high-precision hand tracking, a key component of this system is the HTC VIVE Pro Eye VR Headset for eye-tracking measurements. The Tobii eye tracker integrated within the HTC VIVE. It can be worn with eye glasses and is robust to head movements because it is affixed inside the VR helmet.
Using the HTC Vive Pro, we collected gaze fixations and pupillometry at a rate of 90Hz; this included the left and right pupil diameter and gaze convergence position. Also collected were head position and orientation, relative to the cockpit. Gaze patterns over the course of each maneuver were collected. These are heatmap patterns that are traced from the visual eye path — containing eye fixations. The heatmap of gaze was calculated with a fixed origin relative to the virtual cockpit.

As with the feasibility studies, Chapter 6, to collect photoplethysmography (PPG), electro-dermal activity (EDA), skin surface temperature, and wrist acceleration the Empatica E4 was used (as previously discussed). Data from the simulator including video, gaze, and pupillometry information were recorded. That is, it was possible to review the exact actions and viewpoint taken by the participant while they used the BBXR. The capture software was again utilized, Appendix D. A time-syncing method was employed to ensure timestamp alignment during data reduction. For each subject, timestamps were taken simultaneously on both the BBXR and the E4 data stream. These timestamps were collected
at the beginning and the end of subject data collection. For alignment purposes, a keypress on the server running the BBXR and an iPad interface button were pressed simultaneously to ensure a coarse level of agreement in the timestamps was attained.

7.2. Gaze Data Format

The BBXR provided gaze data at 90Hz, and this included the left and right pupil diameter and the left, right, and combined x-position and y-position on a given field of view (FOV), which was projected onto a two-dimensional square reference frame also known as a screenspace. Three screenspaces were established with 60° FOV and with the same camera origin located just above the pilot’s chair. This position did not change for the duration of the data collection effort.

The three screenspaces were located relative to the cockpit. One screenspace is center and in-front of the multi-function displays and heads-up display. The other two screenspaces were positioned parallel to each other. They were aft and perpendicular to the center screenspace. One is on the left-hand side of the cockpit, and the other is on the right-hand side, effectively surrounding the pilot within the cockpit.

Gaze position was reported for each screenspace. If the subject was looking at something in a given FOV, the position for that screenspace was reported between 0 and 1 for both x and y coordinates. The capture resolution for a given screenspace is 8192 by 8192.

7.3. Data Collection

The data collection effort consisted of a repeated measures experiment. Each flight maneuver was flown at least twice. The experiments were approved by a university IRB, application H18-105-LARE. The forty test subjects consisted of twenty-one pilots, nine operators, and ten novices. The individuals within pilot group represented diverse flight backgrounds, all had flying experience in heavy, rotary, and/or fighter-type aircraft. Some individuals within the pilot group had commercial flying experience. Operators included naval flight officers (NFOs), combat systems officers (CSOs), remotely piloted aircraft (RPA) sensor operators,
and avionics technicians; all with some flight or simulation experience. Novices consisted of those who had no aircraft experience at all. Gaze data and screen capture video were recorded for each maneuver.

The maneuvers flown during this experiment are listed as follows:

- **Cruise Maneuver**: the subject was instructed to fly straight and level maintaining 12,500 ft and 350 knots-indicated airspeed (KIAS) with tolerances of ±100 ft and ±15 KIAS for five minutes.

- **Normal Takeoff**: the subject was positioned on the centerline of the runway 13R at (simulated) Falon Naval Air Station (NAS) KNFL. The subject was asked to smoothly apply max power, rotate at 140 KIAS, and pitch between seven and ten degrees nose high – climbing 3,000 ft and leveling off. Tolerances included: ±1°, ±10 ft centerline, and ±2 deg runway heading.

- **Normal Landing**: the subject was positioned on final to runway 13R at Falon NAS. At decision height, 500 ft above ground level (AGL), the subject initiated a full-stop landing. The subject was instructed to verbalize his or her desired touchdown point upon nearing the runway. Tolerances included the nose-wheel within 10 feet of centerline.

- **Boundary Avoidance Tracking — Longitudinal Axis**: the subject was positioned behind a target aircraft that moved periodically, at random, in the vertical axis. As the target aircraft moved, the subject was asked to keep their aircraft’s crosshair inside of a defined boundary about the target’s longitudinal cross-section. With the progression of the maneuver, in a build-up manner, the task difficulty was increased by reducing the boundary spacing.

- **Boundary Avoidance Tracking — Lateral Axis**: the subject joined on a target aircraft’s right-wing. The target moved periodically and at random intervals in the vertical axis. As the target aircraft moved, the subject was asked to keep their aircraft’s wing or canopy handle inside of a defined boundary about the target’s lateral cross-section. With the progression of the maneuver, in a build-up manner, the task difficulty was increased by reducing the boundary spacing.
- **Air Intercept:** the subject begin flying straight and level. The subject obtained a radar lock on bandit aircraft. They offset his/her aircraft 30° left or right and descend 10° nose low to the bandits altitude. At level-off, the pilot accelerated to greater than 400 KIAS, and executed an intercept/escort profile. Subject closed for visual identification (VID) and verification of the aircraft markings (fin flash).

![Air Intercept Diagram](image)

Figure 7.2: Air Intercept Diagram

For both boundary avoidance tracking (BAT) tasks, when the subject overshot a boundary, they were asked to rapidly recover and place the aircraft back into position and continue the maneuver. The BAT tasks were designed to increase the required pilot workload to complete the maneuver. To this end, the simulator operator had the ability to manipulate the pitch control laws to increase/decrease the overall response of the aircraft. The operator altered the control laws in such a manner as to ensure the subject was stabilized before stepping to the next adjustment in a buildup manner. In practice, this ability proved more useful in compensating for inceptor input (stick) deadband than inducing workload. The subject flew each BAT maneuver for a minimum of five minutes.

In support of SAGAT, Section 4.1, during the air intercept scenarios at the descending pull for offset and roughly 5 NM from the bandit, the simulation was frozen and the screens blanked out. A random subset of questions pertaining to the state of the situation in the simulation environment were asked. Upon answering all questions, the simulation was continued. Appendix C contains the randomized lists of questions.
Prior to data-collection each subject was given five minutes to familiarize themselves with the aircraft, during which any questions were answered. For Novices, they were shown how to use the stick, throttle, rudder pedals, and the heads-up-display (HUD) symbology was explained. During each maneuver the subject was asked to perform the maneuver as he or she normally would. Novices were asked to perform the best they could, and no guidance was provided on proper gaze or scanning patterns. Given the air intercept task is a more complicated maneuver each subject was given the opportunity to practice the maneuver once; this demonstration data was also captured.

7.3.1. Demographics

Table 7.1 lists the subject demographic information over forty test subjects. This includes the sample mean, standard deviation, min and max values for subject age, flight hours, and number of aircraft flown. Further, the percentages for type of flight experience are listed. Note that the flight experience is not exclusive to military or civilian, a subject can have both. Therefore flight experience can add up to greater than 100%.

<table>
<thead>
<tr>
<th>Subjects</th>
<th>Age</th>
<th>Hours</th>
<th>Aircraft Flown</th>
<th>Flight Experience</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(\bar{x})</td>
<td>Std Dev</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>All Subjects</td>
<td>42.0</td>
<td>10.7</td>
<td>22</td>
<td>61</td>
</tr>
<tr>
<td>Pilots</td>
<td>47.4</td>
<td>7.3</td>
<td>35</td>
<td>61</td>
</tr>
<tr>
<td>Operators *</td>
<td>37.9</td>
<td>20.0</td>
<td>23</td>
<td>43</td>
</tr>
<tr>
<td>Novices</td>
<td>34.8</td>
<td>22.8</td>
<td>22</td>
<td>53</td>
</tr>
</tbody>
</table>

*For operators, operator time and pilot time, if applicable, are reported combined.
7.4. Conclusions

This full study was conducted during the summer of 2019. The data were collected from 11 June 2019 to 28 August 2019. All the tasks in this experiment took around 2+ hours to complete per subject. The temperature in the simulator room was set to a constant 68°F for each subject’s data collection. We built a data set covering all 40 subjects including 21 pilots, 9 operators, and 10 novices. Over 40 hours of physiological maneuver data were collected inclusive of the time between maneuvers with an average of twelve maneuvers per subject. The data collected with the methodology described in this chapter was in support of the overall objective; and it would be used to create classifiers needed that can reliably measure cognitive load, gaze scan patterns, and situational awareness using wearable and environmental, physiological sensors.
8.1. Introduction and Motivation

Context-aware computing is an operative force that facilitates ubiquitous computing. Through context, a machine can be rendered capable of response, projection, and adjustment in support of the human-in-the-loop, which increases the “[informational] bandwidth” [61] of an interaction between a human and a machine. The value of understanding the context of the internal state of the human user is not to be understated. By understanding these internal states and granting a machine access to this context we increase the informational bandwidth and enrich the interaction between the human and the machine. With this, several ideas have emerged: (1) Acceleration and amplification of knowledge acquisition and understanding of an exercise or skill through insights into the moment of skill mastery; (2) the relation of operational or academic student successes to their collected human performance data, which can shed light on the understanding of a subject’s susceptibility to different types of error that might be inferred; (3) real-time, self-assessment and personal development through feedback systems for a subject outside of student environments; and (4) adaptive user interfaces adjusting themselves based upon the inferred internal state of the user — situational awareness, attention, cognitive load, emotion, stress, engagement, etc.

Within the UbiComp community Yvonne Rogers highlighted augmented learning environments as a key application area within the UbiComp community, further describing the classroom is an under-explored area in UbiComp research [204]. Moreover, objectively measuring internal states such as cognitive load has “significant implications for adaptive automation” [43], such as adaptive aiding or adaptive task allocation [209]. Haapalainen et al. underscored the value in understanding and measuring such latent human metrics.
in order to discern the best opportunity to “proactively and seamlessly [provide] the right information at the right time” [104]. This motivates the need for context related research in the UbiComp community that focuses on specific application areas, where context can be defined more concretely or in areas where context has a stricter definition. In this research, we focus on the application area of aviation.

In aviation, context- or situation-aware avionics — capable of assisting either the flight instructor or the aviator in understanding the human-machine state [36] in real-time — is highly sought after. Instructors seek to understand the internal context of students while they learn. This is important in assessing their development of many diverse skills. Performance of aviators is currently assessed through instructor observation and review. Through this observation instructors can intuit the internal context of aviators while they learn. While understanding this latent context is considered a key (but challenging) assessment, the automated sensing of this context has not fully materialized. By inferring learner context automatically, instructors and trainees can more easily determine when mastery of a particular skill is achieved, expediting the learning process and, potentially, improving the overall quality of learner understanding.

We seek to measure human performance objectively. To this end, a way of objectively measuring cognitive load is needed. There are many works that have endeavored to provide a solution to this problem, until now no one has shown significant classification accuracy across all subjects [4, 8, 43, 82, 104, 138, 154, 194]. Our contribution is a generalizable model that works inter-subject and inter-task.

In this research we use the aviation domain for inducing and evaluating cognitive workload. The methods and applications used within this research are applicable to, and may be replicated within, an array of other domains or tasks such as driving an automobile. The activities of this research involved the collection of physiological measures relevant to aviators as they undergo simulated flight scenarios for both low and high levels of induced workload. We propose two versions of a new method for cognitive load classification: one with fixed-width input, and the other variable-width input. The basic structure of these
models take shape through the use of X-Vector-like architectures and deep, multi-modal, multi-task, and generative learning. Multiple 1-D modalities from physiological sensors are encoded into a latent space. Statistics layers are calculated and passed through to the bottleneck, and finally through to multiple task layers. The task labels are derived from subjective rating scales — both NASA-TLX and Bedford workload were collected. Data are collected in a mixed-reality training environment using a physical flight simulator, a virtual reality environment, and a gaze-tracking sensor for collecting pupillometry and a wrist-worn sensor suite for capturing photoplethysmography, wrist acceleration, peripheral skin temperature, and electrodermal activity. From these devices the multiple synchronized data streams are classified into cognitive load. These results are compared to a more traditional classifier — random forests. We detail our work as follows:

- We designed and carried out a human subjects experiment for aviators in a variety of flight scenarios, discussed in Section 9.2.5.
- We propose methods for subjective label alignment, discussed in Section 8.2
- We investigate three competing architectures, a random forest and two deep learning classifiers, and we show accuracy across subject, subject-type, and maneuver.
- We present our method’s generalizability through test and evaluation of the models with real-flight data using five test pilot subjects.

Understanding the internal context of an aviator during a simulation are important aspects of the learning experience. With an objective cognizance of human cognition, it may be possible to personalize the learning experience for students, expediting the educational effectiveness by maximally challenging the learner without overloading them.
8.2. Data Reduction

Despite the careful design of our human subjects tests, a number of noise sources are unavoidable. To mitigate the effects of noise (both intra- and inter-subject) the workload labels were manipulated in three ways: (1) inter-subject alignment, (2) label correction, and (3) label imputation. We explain the reasoning and methods of correction for each of the three methods in the following sections. We consider our methods of label alignment to be a novel contributions of this work and, therefore, elaborate on this method in detail. Finally, we conclude this section with discussion of preprocessing and standardization of the sensor data so that it is appropriate for use in a classification model.

8.2.1. Inter-subject Alignment

A key issue with the classification of cognitive load is proper labeling. An absolute inter-subject scale does not exist, but is the ultimate goal of many cognitive load studies. As previously discussed (Section 4.3), using the assumed cognitive demand of the task or the mental load characteristic (e.g., the estimated difficulty of a task) is limited. Any two subjects can potentially accomplish the same task with the same level of performance using varying levels of requisite cognitive load. This is often dependent on some level of their skill at a task, especially for learned (automated) schemas that bypasses working memory like playing an instrument \[45,179,252\]. Chen et al. [43], provided an example of the variance of subjective labels when compared to the validated task difficulty finding the subjective labels to be the best discriminator. But this only works intra-subject; it breaks when we attempt to generalize the method to be inter-subject. Haapalainen et al. and Appel et al. attempted such a feat, but Haapalainen et al. did not succeed and Appel et al. was left with mixed results \[8,104\].

We argue that the mixed results of Appel et al. are not surprising because there is a fundamental problem with using the task difficulty as the label. For example, let us assume for a moment that an absolute scale for cognitive workload exists — a scale from one to ten. We can define a workload inducing task with two levels, where the second level is more
difficult than the first. Two subjects participate in a repeated measures experiment. Subject one reports an average absolute cognitive load of one on the first level and a cognitive load of six on the second level. Subject two reports an average absolute cognitive load of six on the first task, and a cognitive load of nine on the second level. Both completed the tasks with equal performance. Clearly, subject two required more workload to attain the same level of performance for both levels of the task. Assuming enough quality physiological data were collected, a binary *intra*-subject classifier can be built for each subject using the task difficulty as labels. However, if an *inter*-subject attempt is made, subject data for task one — a one and six on the absolute cognitive load scale — are passed to the model during training with similar difficulty-level labels. And for task two, a six and nine. Thus, this presents a conflict. This is compounded by the fact that the subjective reported cognitive load from each subject, in real life, is not absolute (one subject reporting a cognitive load of “one” could be another subject’s “two”). Because the subjective labels are not absolute, we must find a way to align them between subjects.

In this research we used both Bedford and NASA-TLX rating scales. It is important to note that the Bedford workload scale was intended to be used as a subject- and task-relative workload measure. And it has a tendency towards less sensitivity at the low end of workload ratings. Further, Hart and Staveland did not intend NASA-TLX for “absolute judgments or comparisons across different types of tasks,” finding them not very meaningful given subjects are not likely to remember specific instances of low, medium, or high workload [113]. This could explain the observed phenomenon in Chen et al.’s work [43], where workload comparisons at finer resolution than three levels were no longer in relative agreement.

In our effort to align the subjective labels, we made a critical assumption. We bound our data by assuming we achieved minimum and maximum cognitive load for each subject. Our experiment was designed to acquire both minimally and maximally loaded data from the subjects; therefore, we conclude that any errors are sufficiently mitigated. While this assumption helps to bound the upper and lower cognitive load scores, it does not inform how to transform scores within the bounds for each subject. Therefore, we conducted a non-linear
transformation to align the subjective ratings across subjects. By this, we mean that, for each subject, their subjective ratings were mapped into a their own uniform distribution; this was done by assuming the individual subject’s minimum rating and maximum ratings are equivalent to the minimum and maximum of the uniform distribution — 0 and 1 on a normalized scale. This means that the x-value of \( q^{th} \) quantile is \( q \) away from the minimum to the maximum of x-values. This assumption allows an estimate of the inverse cumulative distribution function to be used for transforming the ratings (with respect to the number of selected quantiles). In our experiments, we used this transformation method to align labels for each classification task. For each classification task, a subjective rating’s corresponding quantile was used as the task label for categorical and binary classes. That is, binary tasks used two quantiles and multi-class tasks used the number of classes. The transformed values from 10 quantiles were used for the ordinal regression task.

One limitation of our alignment procedure is that it does not account for different subjective ratings across flight maneuvers. Hart and Staveland’s observed a lack of meaning in subjective ratings across tasks [113]. While one method for addressing this may be made to bound within a flight task, it is not clear if each flight maneuver constitutes a sufficiently different task (as defined by Hart and Staveland). In our analyses, however, we did not find that such a task-specific bounding was needed to create a reliable classifier. This could be because these tasks were flown within the same time period for a given subject and the subject perceived their load along the same scale. Or rather, one can surmise that it is just a layer of noise within the subjective ratings for a given subject. We note that this noise can be partially mitigated for pilots through proper training. For example, someone professionally trained, such as a test pilot, might offer a less variable set of ratings because the method of self-assessment and subjective rating was included in their instruction.

The level of correction needed in our analyses highlights the critical need for an objective workload measure. Additionally, it was evident with our data that we were dealing with “overall workload” for the flight maneuver, rather than more fine grained measures such as turning or elevating the plane [179]. Any attempts to divide our data into smaller time
segments with the same workload rating yielded a divergent model during training. This is understandable given the test methodology and standard practices of collecting cognitive load, subjectively. Workload labeling is an open research problem that needs further investigation. Even so, we argue that our proposed alignment procedure sufficiently aligns the data inter-subject so that overall workload for the flight maneuver may be assessed.

8.2.2. Label Correction

Once a method for aligning inter-subject was settled upon, an effort was made to address the intra-subject noise. That is, the noise given the subjectivity of human-subjects and their abilities to discern relative levels of cognitive load. Hart and Staveland had found subjects’ abilities to remember specific instances of low, medium, or high workload fleeting [113]. While we did conduct the experiment within roughly a two-hour period for a given subject, there is likely still noise within the data. Anecdotally, the pilot population is particularly susceptible to “under-reporting” cognitive load—reporting a scenario as easier than it truly was for the pilot. To assist with this issue, we adjusted some labels based upon expert review of the maneuver. In order to inform label adjustment, several additional resources of information were captured including screen-capture video of the pilot performing the maneuver (i.e., what the pilot saw) and the annotations from the researcher conducting the experiment. Post data collection, the data for the specific maneuver was analyzed and the recording was reviewed. Given the test conductor’s note and the screen-capture video, if warranted, the label was adjusted. The criteria for adjustment was that something in the observed rating and maneuver was “at odds” with previous observations for the same pilot. This was conducted before any model construction to avoid overfitting. Using expert opinion to correct observed subjective ratings is common practice in a number of fields including management [57], UX [5], and military [54], among many others.

As an example, subject 18’s second lateral BAT maneuver resulted in a modification. The subject’s raw labels included 90-mental demand, 90-physical demand, 90-temporal demand, 50-performance, 90-effort, and 80-frustration for NASA-TLX, where each sub-scale
was based on a 100 point scale, and a Bedford workload rating of four. The subject was visibly overloaded, struggled to fly the task, had a significant change in skin conductance level, numerous skin conductance responses, many significant 15-20 second pupillary swings, and a low blink count. The subject’s Bedford workload rating was changed to a nine in accordance with the Bedford description — “Extremely high workload. No spare capacity. Serious doubts on ability to maintain level of effort” [208]. In the data collected, only the Bedford rating or Mental Demand sub-scale were adjusted. In total, 57 labels were adjusted from a total of 446 ratings.

8.2.3. Label Imputation

In the collected data, there were missing subjective ratings for two different maneuvers flown by two different subjects. The ratings were missing due to transcription errors. As with label correction the physiological data, test conductor notes, and video recordings were reviewed by an expert. Taking this into account we imputed the labels using other maneuver ratings. In both cases the Bedford rating for the maneuver was present, but the NASA-TLX ratings were missing. After expert review, the mental demand sub-scale rating was imputed a rating of 50 because the Bedford rating was five (halfway for either scale). The remaining sub-scales were copied from a similar maneuver.

8.2.4. Rating Domains

Ultimately we chose to classify three subjective ratings or rating domains. Because the semantic meanings behind each are different, it is useful to inform which rating domain works best for classification. Specifically, we classify the average or equal weight NASA-TLX ratings (RTLX), the mental demand sub-scale of NASA-TLX, and Bedford Workload ratings. Figure 8.1 provides swarm plots for the three corrected rating domains and their associated alignment with respect to 10 quantiles as discussed in Section 8.2.1. As can be seen, the alignment procedure tends to spread the ratings across the range 0-1 relatively equally.
Four classifier tasks were utilized from each rating domain, including: (1) binary, (2) two-class categorical, (3) three-class categorical, and (4) regression. Figure 8.2 provides the corresponding label distributions for each domain. As can be seen, the distribution of labels is mostly balanced after alignment.

Figure 8.1: A comparison of corrected rating domains and their respective inter-subject alignment (10 quantiles): (a) Equal Weight NASA-TLX, (b) NASA-TLX Mental Demand sub-scale, and (c) Bedford Workload
8.2.5. Modality Preprocessing and Standardization

8.2.5.1. Preprocessing

Preprocessing was conducted on both EDA and pupillometry prior to use. For EDA, the tonic and phasic components were separated from the original EDA waveform by using a finite impulse response (FIR) filter. Because the goal of tonic and phasic separation is to separate the high frequency and low frequency components, filtering is a common choice, and we report the exact method of filtering here for reproducibility. An FIR filter was applied both forward and backward, yielding the phasic waveform — this is known as zero phase filtering [103]. Tonic was then calculated by subtracting the phasic waveform from the original EDA waveform. The filter coefficients employed were calculated from a Butterworth filter [34] with two Hz cutoff. The Butterworth filter design was achieved using a least squares estimate of the pass and stop bands. In our analysis, this achieved minimal
pass band ripple effects with sufficient damping. The coefficients were applied forward and backward to intensify the effect as well as remove any delay from the filtering process.

For pupillometry it was necessary to reduce the noise of the signal by eliminating high frequencies and removing the trend line. We again used zero phase filtering [103] and filter out the very high frequencies. Next we detrended, removing the nonzero mean and other trend terms [278]. The trend was calculated by using a Savitsky-Golay filter as a high pass filter. This type of filter approximates the signal using a relatively low order polynomial approximation and has a unique property of keeping the underlying “shape” of the signal (that is, preserving the maxima and minima of the signal, as was its original intent) while removing noise [218]. In our case, we used a filter length of 11 and third order polynomial approximation. The trend was then subtracted from the filtered signal, yielding the detrended signal. After these modalities were filtered, all data were standardized.

8.2.5.2. Standardization

For all three models the modalities were standardized to a zero mean and a standard deviation of one with respect to the segment width of the sample input. For the random forest and variable input deep learning models the segment width and sample width are always the same. However, for the fixed-length architecture, the sample was decomposed into a number of fixed width segments. Standardizing in this way for all models increases the dynamic range and flexibility of the model, enabling a more accurate classifier. Moreover, by standardizing across input segments before being analyzed by the model, the model is more sensitive to relative changes. As with temperature rate of change, this may “ameliorate the effect of individual difference inherent in all physiological parameters” [109]. While this was a subtle change, doing this in lieu of standardizing over the whole dataset (or even across each subject) yielded a significant increase in performance. Thus, we stress the importance of this standardization technique for processing biometric data such as ours—the absolute changes in the signal are not as usable for classification as the relative changes when employing a deep learning architecture.
8.2.6. Engineered Features

While we relied heavily on deep learning for feature extraction, we also engineered more traditional features that capture specific properties of certain modalities. Moreover, these features provided a baseline for comparing the more automated feature extraction used by convolutional networks. For each signal, we calculated the minimum, maximum, standard deviation, and mean. We performed this calculation for the raw EDA signal, peripheral skin temperature, acceleration magnitude, skin conductance response (SCR), interbeat interval, and left and right pupillary response, resulting in \((7 \text{ signals}) \times (4 \text{ aggregations}) = 28 \text{ features}\). These aggregations were calculated in order to “snapshot” the signal’s characteristics in a way that is easily analyzed by a traditional machine learning approach, such as a random forest. Furthermore, we captured more signal specific measures that are common in the biometric analysis community. Specifically, we measured SDRR, pNN50, SCR count, left and right blink, and average blink rate (6 additional features). The pNN50 and SDRR were calculated according to [231]. Specifically, SDRR is the standard deviation of RR intervals, and pNN50 is the percentage of successive RR intervals that are different by more than 50ms, whereby “RR intervals [are] interbeat intervals between all successive heartbeats” [231]. Ultimately, all engineered features were normalized across the dataset between 0 and 1. Finally, most engineered features employed a sliding window approach, whereby peaks and troughs were calculated within each window, and then as estimates for a number of features such as blinks.

With regards to pupillary response, a 20-second window was used to detrended pupil diameter, unlike in [271] where five seconds were used for a baseline. We used the average of the two preceding troughs, which is a much larger baseline of 20s. This was done because we do not know exactly where the external stimuli begins or ends. Further, we did not use the percentage of the pupillary response — \(\frac{\text{peakvalue} - \overline{\text{baseline}}}{\overline{\text{baseline}}}\) — as with [137] nor a correction factor for lighting [189, 271]. The advantages of a percentage pupillary change are understood as providing a standardization regardless of a unit measurement or original baseline size. However, we found it beneficial to take the minimum, maximum, mean, and
standard deviation of the samples, followed by a normalization, between 0 and 1, for the whole dataset. However, This approach may not generalize outside of our application—we benefited from not needing to use a correction factor as the lighting within the HTC VIVE Pro Headset was relatively constant.

For **blinks**, a window size of 1/2 second was used to capture the high frequency blips from the unmodified pupil diameter data. If the left or right eye differed significantly with one eye providing unrealistic results, the other eye was utilized in place of this eye blink data. This problem most often occurred with subjects wearing glasses.

**Skin conductance responses** were measured through the use of a 3.25 second sliding window. First, the peak was found; then the preceding trough 3 seconds prior was found. If the difference is greater than .01 µS, the SCR was recorded and associated statistics calculated. Each sample was normalized between 0 and 1. This was done due to the characteristics of EDA, discussed in Section 5. Ultimately we seek to observe the relative changes in and intensity of SCRs between both samples in lieu of absolutes.

### 8.3. Architecture

In this research three architectures were used to classify cognitive load. Specifically, we used a more traditional random forest model, which served as a comparison for the other two models. And we used two variants of a novel architecture we coin as **Biometric Multi-Modal, Multi-Task, X-Vector architecture** or **BM³TX**. We propose two versions of a new method for cognitive load classification: one with fixed-length input, and the other variable-length input. The basic structure of these models takes shape through the use of X-Vector-inspired architectures and deep, multi-modal, multi-task, and generative learning. As an overview, multiple 1-D modalities from physiological sensors are encoded into a latent space using a variational auto-encoder’s encoder for each modality. Then, taking inspiration from X-Vectors [240], statistics pooling layers are calculated and passed onto the bottleneck (the layers directly following the pooling), and finally through to multiple task layers. In this section, we discuss the random forests model, the necessary deep learning background, the
generative models — variational auto-encoders architectures — used in both versions of the \(BM^3TX\) architecture, and finally fixed-length and variable-length input architectures.

8.3.1. Random Forests

As a baseline, a random forests classifier [26] was built for each task using only the engineered features, as discussed in Section 8.2.6. We assume the reader is familiar with the basic concepts of random forests, and therefore list the relevant parameters used in training. For each model, we employed bootstrapping with sampling preference proportional to mis-classified training samples, randomized bagging of features based on \(\sqrt{N_f}\), balanced class weights (inversely proportional to frequency in training set), and 1,000 trees. The split criterion for both the binary and two-class categorical classifiers used the gini impurity [26], while the regressor versions used mean squared error.

8.3.1.1. Training

Generically, training was conducted in the same manner for all three architectures. Cross-subject k-fold cross-validation was utilized and folds were stratified across participant background: novice, operator, and pilot. Because there were only nine operators, the lowest subject count (Chapter 7), the stratification of folds across subject type was only possible for eight training folds. This left a ninth grouping of novice, operator, and pilot for the test fold. Practically, this meant that each fold had at least one example of a novice, operator, and pilot. It is important to note that by using-across, subject cross validation, we ensure that the model is built assuming no calibration biometric data for the test participants are available. For the random forests classifier each task was trained as a separate model.

8.3.2. Deep Learning Background

Both versions of the \(BM^3TX\) architectures are similar in that they follow the basic structure through the use of multiple modality variational encoders, statistics layers, bottleneck layers, and multi-task layers. They take advantage of several key concepts from within ma-
chine learning, Chapter 3: (1) multi-task learning, (2) multi-modal learning, (3) variational autoencoders, and (4) x-vectors. Further, the variable-length input version takes advantage of transfer learning. We discuss each in turn. In our discussion of neural networks, we adopt the terminology of “dense layers” to refer to fully connected layers followed by a non-linear activation function and convolutional layers denote 1-D temporal convolutions. Moreover, we use the term “separable convolution” to denote convolutional layers where each channel is separated and convolved with a different filter before it is convolved with a second filter that weights the outputs of each filter. This differs slightly from the definitions of “separable convolution” used in the signal processing community, where a higher dimensional convolution can be carried out by multiple chained lower dimensional convolutions (though in practice, the separable convolution can approximate this behavior).

Figure 8.3: Example PPG variational autoencoder architecture
8.3.3. Modality Variational Autoencoders

We used VAEs as a preprocessing step in encoding each modality. That is, we built six VAE models (one for each modality) - PPG, temperature, wrist acceleration magnitude, raw EDA, the tonic component of EDA, and detrended pupillary response. An attempt was made to build a VAE for the phasic component of the raw EDA signal, but the model was unable to find the gradient (despite numerous attempts using gradient stabilization techniques). The hyperparameters for each model can be found in Table 8.1. These parameters were chosen because they produced the best loss functions on held out sets (that is, the lowest reconstruction errors and most normally distributed latent space samples). Figure 8.3 provides a visual example of these VAE architectures.

Table 8.1: VAE Architecture: The VAE configurations for each modality. A layer’s parameters are denoted as ‘Layer Type < KernelSize > – < Strides > – < Filters > – <Activation Type (ReLu if not noted)].’

<table>
<thead>
<tr>
<th>Photoplethysmography</th>
<th>Wrist Acceleration</th>
<th>Peripheral Skin Temperature</th>
<th>Raw EDA</th>
<th>Tonic</th>
<th>Pupillary Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Length</td>
<td>98</td>
<td>32</td>
<td>98</td>
<td>98</td>
<td>98</td>
</tr>
<tr>
<td>Encoder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Conv1D 3-1-128</td>
<td>Conv1D 3-1-128</td>
<td>Conv1D 3-1-128</td>
<td>Conv1D 3-1-128</td>
<td>Conv1D 3-1-128</td>
<td>Conv1D 3-1-128</td>
</tr>
<tr>
<td>SepConv1D 3-2-64</td>
<td>SepConv1D 3-1-64</td>
<td>SepConv1D 3-2-64</td>
<td>SepConv1D 3-2-64</td>
<td>SepConv1D 3-2-64</td>
<td>SepConv1D 3-2-64</td>
</tr>
<tr>
<td>SepConv1D 3-2-32</td>
<td>SepConv1D 3-2-32</td>
<td>SepConv1D 3-2-32</td>
<td>SepConv1D 3-2-32</td>
<td>SepConv1D 3-2-32</td>
<td>SepConv1D 3-2-32</td>
</tr>
<tr>
<td>SepConv1D 3-2-16-tanh</td>
<td>Weighted Sum*</td>
<td>SepConv1D 3-2-16-tanh</td>
<td>SepConv1D 3-2-16-tanh</td>
<td>SepConv1D 3-2-16-tanh</td>
<td>SepConv1D 3-2-16-tanh</td>
</tr>
<tr>
<td>Weighted Sum*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Latent Dimensions (Mean and Std. Dev. - Dense Layers)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>16</td>
<td>9</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>Decoder</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dense-224</td>
<td>Dense-320</td>
<td>Dense-224</td>
<td>Dense-224</td>
<td>Dense-224</td>
<td>Dense-70,656</td>
</tr>
<tr>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
</tr>
<tr>
<td>SepConv1D 3-1-16</td>
<td>SepConv1D 3-1-16</td>
<td>SepConv1D 3-1-16</td>
<td>SepConv1D 3-1-16</td>
<td>SepConv1D 3-1-16</td>
<td>SepConv1D 3-1-16</td>
</tr>
<tr>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
</tr>
<tr>
<td>SepConv1D 3-1-32</td>
<td>SepConv1D 3-1-32</td>
<td>SepConv1D 3-1-32</td>
<td>SepConv1D 3-1-32</td>
<td>SepConv1D 3-1-32</td>
<td>SepConv1D 3-1-32</td>
</tr>
<tr>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
<td>2 x UpSampling</td>
</tr>
<tr>
<td>SepConv1D 3-1-64</td>
<td>Conv1D 3-1-1-linear</td>
<td>SepConv1D 3-1-64</td>
<td>SepConv1D 3-1-64</td>
<td>SepConv1D 3-1-64</td>
<td>SepConv1D 3-1-64</td>
</tr>
<tr>
<td>Conv1D 3-1-1-linear</td>
<td></td>
<td>Conv1D 3-1-1-linear</td>
<td>Conv1D 3-1-1-linear</td>
<td>Conv1D 3-1-1-linear</td>
<td>Conv1D 3-1-1-linear</td>
</tr>
<tr>
<td>Output Length</td>
<td>98</td>
<td>32</td>
<td>98</td>
<td>98</td>
<td>2205</td>
</tr>
</tbody>
</table>

*aA weighted sum across filters i.e. SepConv1D 1-1-1 (BOLD) Pruning location for variable-length BM³TX architecture.


### 8.3.3.1. VAE Training

For training each VAE and for a given modality, the data for each subject were combined and then segmented into the specific input lengths given in Table 8.1, with a 20% overlap from the previous segment. Before training, we randomly separated 10% of the modality data for a final test set. We used the ADAM stochastic optimization method [135] with a learning rate of .0001 for all modalities except for EDA and tonic, which were .01. Standard beta values were used with gradient clipping range of 0.01 to 0.5. We chose the best hyperparameters based on the test results because we were trying to reconstruct the best modality segment irrespective of the subject. With batch sizes of 64 to 16,384 samples, each model was trained for roughly 800 to 2600 epochs, given early stopping criteria on the held out set. Table 8.2 reports the test dataset results. While all modalities are different—especially with respect to their dynamic ranges—each VAE had a strong correlation coefficient of no less than .68. As such, we concluded that the VAE architectures were sufficiently trained and could be used for further processing.

### 8.3.4. Fixed-Length Input BM³TX

The fundamental difference between the two versions of BM³TX focuses on how the modality data is input into the model and processed at the merge layer. Specifically with fixed-length input, the data is segmented to the input size of each modality’s VAE, and passed through their respective VAE (encoder only). Theses latent vectors from the VAEs are then stacked temporally before passing through a statistics pooling layer where mean, standard deviation, minimum, and maximum along the temporal dimension are calculated. It follows that the input modality data must be aligned. In this way, the same coverage
of time for each modality is passed through, regardless of sampling rate. We note that the term “fixed-length” refers to the fact that fixed length VAE outputs are aggregated. The model can process variable length inputs, but does so by stacking overlapping VAE output windows.

8.3.4.1. Modality Data Alignment

To accomplish the needed sample rate alignment a basic unit of time was selected that both covered a sufficiently large amount of time (given a maneuver with overall workload), and one that was divisible by all sampling rates for even coverage of all modalities. In this regard, we chose windows that were 49 seconds in length. Because we actively attempted to induce high workload through a build-up approach, it is assumed that the highest amount of cognitive load was captured towards the end of the data sequence. Therefore, we right justify (temporally) the maneuver data in 49-second chunks, starting at the end of the maneuver and dropping the first several seconds, [0 to 49) seconds, of data at the onset of the maneuver. Table 8.3 provides a description of the alignment parameters given a modality.

Table 8.3: Modality Alignment Parameters

<table>
<thead>
<tr>
<th>Modality</th>
<th>Segment Length</th>
<th>Sampling Rate</th>
<th>Segment Samples</th>
<th>VAE Input Length</th>
<th>Latent Vectors per Segment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Photoplethysmography</td>
<td>49 Seconds</td>
<td>64</td>
<td>3136</td>
<td>98</td>
<td>32</td>
</tr>
<tr>
<td>Wrist Acceleration</td>
<td>49 Seconds</td>
<td>32</td>
<td>1568</td>
<td>32</td>
<td>49</td>
</tr>
<tr>
<td>Peripheral Skin Temperature</td>
<td>49 Seconds</td>
<td>4</td>
<td>196</td>
<td>98</td>
<td>2</td>
</tr>
<tr>
<td>Raw EDA</td>
<td>49 Seconds</td>
<td>4</td>
<td>196</td>
<td>98</td>
<td>2</td>
</tr>
<tr>
<td>Tonic</td>
<td>49 Seconds</td>
<td>4</td>
<td>196</td>
<td>98</td>
<td>2</td>
</tr>
<tr>
<td>Pupillary Response</td>
<td>49 Seconds</td>
<td>90</td>
<td>4410</td>
<td>2205</td>
<td>2</td>
</tr>
</tbody>
</table>

8.3.4.2. Fixed-Length Input BM³TX Architecture

Fixed-Length Input BM³TX architecture consists of the modality, variational encoders, a statistics pooling layer that calculated the mean, standard deviation, minimum, and maximum of the stacked latent vectors. These statistics are then concatenated together and passed through a bottleneck of three dense layers that increase in size — 64, 128, and 256.
The bottleneck connected to four separate tasks, where each task contained two task layers, of size 512 and 256, and an output layer. Specifically, binary, two-class categorical, three-class categorical, and regression were used for each rating domain, see table 8.4. To help mitigate over fitting, each task layer used a dropout rate of 0.5 [245].

Table 8.4: $BM^3TX$ Architectures: A layer’s parameters are denoted as ‘$<$ Dense Nodes $>$ $-$ [Activation Type (ReLu if not noted)].’

<table>
<thead>
<tr>
<th>Fixed-Length Input $BM^3TX$</th>
<th>Variable-Length Input $BM^3TX$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combinations of Modality Variational Encoders</td>
<td>Combinations of Modality Pruned VAE Filter Outputs</td>
</tr>
<tr>
<td>Stacked VAE Latent Vectors</td>
<td>See Table 8.2</td>
</tr>
<tr>
<td>Statistics Layer — Mean, Standard Deviation, Minimum, and Maximum; Flattened</td>
<td></td>
</tr>
<tr>
<td>Bottle Neck Dense Layers</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>128</td>
<td>32</td>
</tr>
<tr>
<td>256</td>
<td></td>
</tr>
<tr>
<td>Multi-Task Dense Layers</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>512</td>
</tr>
<tr>
<td>256</td>
<td>256</td>
</tr>
<tr>
<td>Output Layers</td>
<td></td>
</tr>
<tr>
<td>1-Sigmoid</td>
<td>2-Softmax</td>
</tr>
</tbody>
</table>

8.3.4.3. Fixed-Length $BM^3TX$ Training

Training consisted of cross-subject, folded cross-validation where each fold was stratified across subject type. This equated to eight folds, leaving a ninth for the test dataset. The fold count was driven by the fact that only nine operators participated in the experiment. It is important to note that during training it had become apparent several local minima existed throughout the gradient. In some cases, training would end with a poor result. Therefore, for each fold the best of five training runs was taken.

Training was conducted in three phases. First, the VAEs were trained as discussed in Section 8.3.3. Second, the modality data was passed through the modality encoders and the corresponding statistics were pooled for each subject’s maneuver. This transformed dataset was then parsed into folds for training and testing. The remaining portion of the architecture, from the merge layer through task output, was trained using the ADAM stochastic
optimization method [135] with a learning rate of .0001. Standard beta values were used with gradient clipping of 1.0. L1 and L2 regularization were used each with $\lambda = .001$. We employed a batch size of 64 samples. The models were trained for roughly 250 epochs, given early stopping. Our training results are described in Section 8.4. We choose the best hyper-parameters, given our results through stratified, cross-subject, folded training.

8.3.5. Variable-Length Input BM3TX

Ideally, our goal was a model that could handle variable-length input without the need for windowing from the fixed length VAEs. To this end we took advantage of the inherent ability of convolutions within each VAE (given that they can be applied to any size input). That is, we apply the learned convolutional kernel across the entire input data length — scaling the output with the length of the input [95]. Because convolutions both use parameter sharing and learned equivariant representations, convolutions provide “a means of working with inputs of variable size” [95]. Where parameter sharing is the use of the same parameter for more than one function within a model — the kernel. A simple 1-D convolution operation without flipping is given by:

$$Z_j = \sum_m V_{j+m-1}K_m$$

Where $j$ is the index of input $V$ and output $Z$.

And $m$ is the offset of input $V$ and the index of kernel $K$; for one filter.

In a convolutional neural network the kernel is used at nearly every position of the input. Thus, a convolutional layer learns a set of parameters for all locations regardless of input length. Because of this, the layer has the equivariance property; that is, the layer is equivariant to translation. Equivariance means, “if the input changes, the output changes in the same way” — where $f(g(x)) = g(f(x))$ [95]. Goodfellow et al. provided an example for time-
series data. Specifically, a convolution captures different learned features that appear within the input. Should a similar feature appear later in time, that representation will manifest within the output. Therefore, convolutions are robust to translations and are independent of input length, which is exactly what we require given variable length maneuver data. “Convolution[s for] the processing [of] variably sized inputs makes sense only for inputs that have variable size because they contain varying amounts of observation of the same kind of thing — different lengths of recordings over time” [95]. With convolutions, input length is a free parameter.

With the promising results of fixed-length $BM^3TX$, we evolved the architecture by eliminating the weighted summation and dense layers of the trained VAE, focusing only on the encoding convolutional filters. In the same fashion as with x-vectors, Section 3.6, we apply a statistics pooling layer directly to the outputs of the filters from the dimensionality reducing convolutional neural network of the VAE (encoder only). The statistic layer computes the mean, standard deviation, min, and max across the filters, for each modality. With the statistics pooling layer directly attached to the filter output of the convolutional layers, the convolutions are decoupled from the requirement of a global input length. This is because the statistics layer will always produce the same output size, giving the dense bottleneck network a fixed-length input size. However, there is a caveat, which depends on the implementation and optimizations of a given deep learning API, in our case Keras. Each sample in a batch must be the same length; therefore, we used a batch size of one (which can dramatically increase training time because it reduces parallelism in the feed-forward and back propagation protocols).

The new variable-length input $BM^3TX$ architecture thus consists of the modality variational encoders which were pruned just before the separable convolutional layer with the hyperbolic tangent activation. Moreover, the filters were not frozen during training to allow fine tuning. Therefore, the VAE training provided a starting point for the full optimization. The pruning location is highlighted on Table 8.1 in bold. The statistics pooling layer is calculated across each filter output, thus “aggregat[ing...across] the time dimension” [240],
and concatenated. Two bottleneck layers are used, with 64 and 32 nodes. The bottleneck connects to four separate tasks, where each task contained two task layers, of size 32 and 16, with an output layer. Specifically, binary, two-class categorical, three-class categorically, and regression were used for a given rating domain, see Table 8.4.

8.3.5.1. Variable-Length BM3TX Training

As with fixed-length, BM3TX training consisted of a across-subject, folded cross-validation where each fold was stratified across subject type. This equated to eight folds, leaving a ninth for the test dataset. The local minima issue plaguing the fixed-length version continued with this architecture. It was apparent several local minima existed throughout the gradient. In some cases, training would end with poor results. Again, the best of five training runs was taken.

Training was conducted end-to-end with the entire sample passed as a single batch. This architecture was trained using the ADAM stochastic optimization method [135] with a learning rate of .001. Standard beta values were used with gradient clipping of 1.0. The model was trained for 50 epochs. Section 8.4 provides our training results. The best hyper-parameters were chosen given our results through stratified across-subject folded training.

8.3.5.2. Model Training Discussion

Beyond hyper-parameter tuning, there were several forms of regularization attempted to improve the performance of this architecture, including L1, L2, and dropout. L1, L2, and L1 mixed with L2 regularization were not used because they did not improve the model performance. Interestingly, when dropout [245] was attempted a vanishing gradient or an exploding gradient, instability would occur. Similarly, if the data were split into temporally smaller segments to augment the dataset size, the model diverged. As discussed in Section 8.2.1, we hypothesize this is because the subjective ratings reflects the entire maneuver, not smaller segmented time windows. That is, half a maneuver might be subjectively rated as a potentially different level of load, which invalidates this style of data augmentation for
our application. While we did trim the maneuver for the fixed-length BM3TX architecture, Section 8.3.4. The trimming resulted in no more than one sample of maneuver data, and it was representative of the whole maneuver.

Finally, we focus on a discussion of the use transfer learning. In our final version of the architecture we use all layers up to the layers preceding the hyperbolic tangent activated convolutional layer for transfer learning. However, we did not freeze these transferred layers because this flexibility yielded the best performance, albeit it will train a quality model if these transferred layers are frozen. More interestingly, this architecture will train with using a truncated normal set of randomized weights in lieu of the learned weights from the variation encoder. However, this is done with some effort, and it does not perform as well. The learning rate must be greatly reduced, and even then, there is a high probability of an exploding gradient during training. Clipping does not seem to mitigate this divergence. Ultimately, the learned representations from the modality variational encoders were utilized to better inform and accelerate training, while improving accuracy. However, given these layers were left unfrozen after transfer, the property of a learned latent space with an approximate diagonal covariance is no longer assured because the VAE loss is no longer optimized once the layers are transferred.

8.4. Results

There are several ways to configure each architecture while training. For example, architectures can employ a selection of modalities, subsets of tasks and auxiliary tasks, inclusion of subjective rating domain choices, or trained on subsets of similar participants (such as flight experience). Once these design decisions are made, the specific training for each architecture can commence, as discussed in Section 8.3. However, to train models while varying every possible subset and hyper parameter combination is intractable. Instead, we guide the selection of training subsets and parameters by the research question we wish to elucidate.
Table 8.5: Architecture fold metrics for each subjective ratings domain — single model

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Binary</th>
<th>2-Class</th>
<th>3-Class</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc.</td>
<td>TPR</td>
<td>Prec.</td>
<td>AUC</td>
</tr>
<tr>
<td>Random Forests</td>
<td>70.4%</td>
<td>70.4%</td>
<td>70.2%</td>
<td>.782</td>
</tr>
<tr>
<td>Fixed-Length BM^3TX</td>
<td>79.6%</td>
<td>79.6%</td>
<td>79.0%</td>
<td>.861</td>
</tr>
<tr>
<td>Variable-Length BM^3TX</td>
<td>78.9%</td>
<td>78.9%</td>
<td>79.4%</td>
<td>.86</td>
</tr>
</tbody>
</table>

(a) RTLX

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Binary</th>
<th>2-Class</th>
<th>3-Class</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc.</td>
<td>TPR</td>
<td>Prec.</td>
<td>AUC</td>
</tr>
<tr>
<td>Random Forests</td>
<td>69.0%</td>
<td>69.0%</td>
<td>68.9%</td>
<td>.762</td>
</tr>
<tr>
<td>Fixed-Length BM^3TX</td>
<td>79.6%</td>
<td>79.6%</td>
<td>79.8%</td>
<td>.861</td>
</tr>
<tr>
<td>Variable-Length BM^3TX</td>
<td>80.4%</td>
<td>80.4%</td>
<td>80.5%</td>
<td>.867</td>
</tr>
</tbody>
</table>

(b) Mental Demand

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Binary</th>
<th>2-Class</th>
<th>3-Class</th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc.</td>
<td>TPR</td>
<td>Prec.</td>
<td>AUC</td>
</tr>
<tr>
<td>Random Forests</td>
<td>66.7%</td>
<td>66.7%</td>
<td>66.9%</td>
<td>.758</td>
</tr>
<tr>
<td>Fixed-Length BM^3TX</td>
<td>80.7%</td>
<td>80.7%</td>
<td>80.7%</td>
<td>.863</td>
</tr>
<tr>
<td>Variable-Length BM^3TX</td>
<td>80.7%</td>
<td>80.7%</td>
<td>80.8%</td>
<td>.861</td>
</tr>
</tbody>
</table>

(c) Bedford Workload

8.4.1. Initial Models - Multiple Ratings Domain

The first question we ask is: Is there an advantage to using our multi-task architectures compared to random forests? To answer this, we firstly compare performance of random forests against the fixed- and variable-length BM^3TX models, with all modalities employed and all participants included. Table 8.5 lists the performance of each architecture configured with all five modalities including the engineered features. While random forests consist of a model for each task, the two multi-task deep learning models each have a set of tasks that are trained for each ratings domain; specifically, binary, 2-class and 3-class categorical, and regression. The 2-class, 3-class, and regression tasks are intended as auxiliary tasks which boost training for both BM^3TX architectures. The 3-class and regression models were built using individual RFs for comparison. For each task the table lists model accuracy (Acc.), true positive rate (TPR, also known as recall or sensitivity), precision (Prec.), and the area under the curve (AUC) of the receiver operating characteristic, which provides a
good estimate of the trade-off between TPR and false positive rate (FPR). For regression the correlation coefficient (r-value), mean squared error (MSE), mean absolute error (MAE), and explanation of variance are used.

The random forest classifier sets a baseline of \(\approx 70\%\) inter-subject accuracy (RTLX) with near matching TPR and precision — this trend continues with the other models. Both \(BM^3TX\) architectures achieve \(\approx 80\%\) inter-subject accuracy. In all cases the AUC is high respective to accuracy, TPR, and precision. These AUC values — representing a trade-off between TPR and FPR — signify a strong model regardless of threshold. In the case of the fixed-length \(BM^3TX\) architecture, the 2-class categorical task out-performed the binary task. For clarity, the 2-class categorical task has a two-node, softmax output in lieu of the more traditional single node, sigmoid binary output. Because a sigmoid is just a special case of the multi-class softmax, the difference is in the architecture, the flexibility provided by two output nodes and accompanying weights vs one, and the stochastic nature during the training of the model — this makes the 2-class categorical task an acceptable auxiliary task. Should it outperform the binary variant, there is no reason not to use the 2-class categorical task for classification, as it would behave operationally the same.

The 3-class and regression tasks did not perform strongly. However, they may have provided a useful service as auxiliary tasks during each architecture’s multi-modal, multi-task training (we explore this more in a later comparison). Figure 8.4, provides BlandAltman plots for regression. On average, across the eight models of eight folds there is a tendency to be within \(0.5\) (raw confusions). The mean is close to zero, an indication of low bias, but it may have a slightly biased homoskedasticity, as indicated by the unequal variance across values. Clearly there is tighter variance near the extrema. Unfortunately the 95\% CI is rather high. However, most points fall within the second and third quartile, and they are less than \(\pm 0.22\). The test set and single fold validation set examples are also interesting. They are more homoskedastic, displaying a slightly smaller CI of nearly \(0.38\), and the majority of the points are within \(\pm 0.2\).
Thus we conclude that the $BM^3TX$ model performs superior to random forests, and that this performance is meaningful for binary and 2-class categorical classification of CL measures. However, we also conclude that the performance of 3-class and regression for CL is not sufficiently strong to operationalize into a tool or interface. But, it is unclear if these measures provide advantages as auxiliary tasks for enhancing the performance of the binary classification. We now focus on methods for enhancing binary classification and investigate if there are advantages to the multi-domain and multi-class training employed.

Figure 8.4: Variable-Length $BM^3TX$: BlandAltman plot for regression, including (a) all eight, (b) test set, and (c) one fold
8.4.2. Multiple Domain vs Single Domain Models

Because of the complexity of these deep learning models, given the number of tasks for each ratings domain, it is reasonable to ask: *Is there an advantage to using more than one ratings domain?* To investigate this question, we trained models with and without knowledge from other domains. Table 8.6, lists the results for the variable-length $BM^3TX$ trained on only one, single ratings domain — RTLX, mental demand (MD), or Bedford workload (BED), respectively — with all modalities and hand-engineered features. Exploring these results, regression performance improved and binary performance increased slightly, all > 81.0%, which were attained for all three models. When visually compared to the multi-domain models, there is an apparent improvement. But is there a significant difference between single domain and multi-domain models?

To answer this question we used the McNemar’s test [158], or more specifically the continuity corrected version [73]. We aggregated every fold of the single domain variable-length $BM^3TX$ models, Table 8.6, and the multi-domain model, Table 8.5, both of which used all modalities as input training features. From these groups, we compared them statistically to understand if there is a significant benefit to using multiple domains for training or training within each domain. We compare multi-domain training to single domain training using a McNemar test of the binary classification outputs and conclude that the difference is not significant (Ndg=36, RTLX: $p=.24$, MD: $p=.61$, BED: $p=.88$) (where Ndg is the off diagonal elements of the McNemar contingency table). We further employed the use of an F-test of equal variance to the residuals of the regression output with similar conclusions (N=270, RTLX: $p=.17$, MD: $p=.13$, BED: $p=.90$) (although the Bedford output data is not sufficiently distributed normally according to a Shapiro-Wilkes test [232], which limits the power of the F-test of residual variance for this output). Because the performance difference is not significant in training all domains versus a single domain, we conclude that there is no need to employ effort in synchronizing data for a multi-domain model. With the more simplified single-domain models, it is then reasonable to ask the question: *Is there a performance difference between a single-task model and a multi-task within the same ratings domain?*
Table 8.6: Architecture fold metrics for variable-length $BM^3TX$ — a separate model for each subjective ratings domain

<table>
<thead>
<tr>
<th>Ratings Domain</th>
<th>Binary</th>
<th></th>
<th></th>
<th>2-Class</th>
<th></th>
<th></th>
<th>3-Class</th>
<th></th>
<th>Regression</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Acc.</td>
<td>TPR</td>
<td>Prec.</td>
<td>AUC</td>
<td>Acc.</td>
<td>TPR</td>
<td>Prec.</td>
<td>AUC</td>
</tr>
<tr>
<td>RTLX</td>
<td>81.9%</td>
<td>81.9%</td>
<td>82.6%</td>
<td>.864</td>
<td>80.7%</td>
<td>80.7%</td>
<td>81.5%</td>
<td>.867</td>
<td>58.5%</td>
</tr>
<tr>
<td>Mental Demand</td>
<td>81.9%</td>
<td>81.9%</td>
<td>82.0%</td>
<td>.88%</td>
<td>82.2%</td>
<td>82.2%</td>
<td>82.3%</td>
<td>.883</td>
<td>59.3%</td>
</tr>
<tr>
<td>Bedford Workload</td>
<td>81.5%</td>
<td>81.5%</td>
<td>81.5%</td>
<td>.859</td>
<td>80.7%</td>
<td>80.7%</td>
<td>80.8%</td>
<td>.852</td>
<td>60.4%</td>
</tr>
</tbody>
</table>

8.4.3. Multi-Task vs Single Task Models

To answer the question of the necessity for multiple tasks within a single domain, we trained several models, given the power-set of the five modalities, with either a single binary task or four tasks (binary, 2-class and 3-class categorical, and regression). That is, excluding the null set, we trained 31 combinations. We further included two more variants for a total of 33 models per ratings domain and task training type (single- or multi-task). The two variants were the four modalities of the Empatica E4 and the combination of all modalities, i.e., the addition of pupillary response. We combined these two versions with their corresponding engineered features that are paired with their respective modalities. We call these sets a ratings domain family of models. In all, 192 models were built and trained, half single-task and half multi-task. By taking the average accuracy across each domain family of models and plotting, there is a clear difference between single-task and multi-task, as depicted in Figure 8.5. In fact, the single-task models do not train beyond apparent chance. We then aggregated the output of every fold for every model within a domain family of models and task training type, which created two large groups of data. That is, the multi-task outputs and the single-task outputs. We then compared them statistically to understand if there is a significant difference between the performance of the two groups as a whole. We calculated the McNemar test of the binary classification between outputs of each group, single-task as compared to multi-task, finding a p-value < 0.001 for all tasks and we conclude a significant difference exists. This signifies that, while we only need one ratings domain for a quality model, we still need multiple tasks within that domain for training. Therefore, we next ask the question: Does one domain have superior performance across tasks over another?
8.4.4. Superior Domain Family

In order to elucidate which family, if any, has superior performance we again aggregated every fold for every model within a single domain family of models, as with Section 8.4.3, with respect to the variable-length $BM^3TX$ multi-task architecture. Figure 8.6 depicts the accuracies of each model of each family. We compared domains by calculating the McNemar test of the binary classification outputs comparing each domain in turn for a total of three tests. We conclude that the difference is not significant between equal weight RTLX and mental demand families (Ndg=2161, Rat-TLX vs Mental Demand:p-value=.763) (where Ndg is the off diagonal elements of the McNemar contingency table). However, for both RTLX and mental demand when each are compared to Bedford, the difference is significant (Ndg=2526, RTLX vs Bedford:p-value=.003) and (Ndg=2399, Mental Demand vs Bedford:p-value=4.12e-07). Given, there is not a significant difference between RTLX and Mental Demand domains, and both are superior in performance to the Bedford workload domain (Table 8.6), we choose equal-weights RTLX in lieu of mental demand for simplicity. This is done in the hope that it will generalize better when more abundant and unobserved data becomes available. The creators of NASA-TLX implemented multiple sub-scales to provide clarity when assessing cognitive load [113].
With a superior domain family chosen, we ask how models compare within the RTLX family of models. Specifically, how does the number of modalities and participant training influence the performance of variable-length $BM^{3}TX$ multi-task models? However, this elicits two related, but distinct research questions: (1) *Is the performance of the architecture within a domain influenced by the combinations of input modalities?* and (2) *Is the performance of the architecture within a domain influenced by having been trained on participants with similar experiences?*
8.4.5. Modality Combination Performance

Given the various combinations of modalities, in order to explore the variance in performance, we trained an RTLX family of multi-task models in the same manner as in Section 8.4.3. The freedom provided by the number of modalities as a hyper-parameter can be keenly observed on Figure 8.7. Even with only three modalities, performance above 80% is attainable, and with two modalities, performance is about 75%. And for the case of the 2-Class categorical task, by itself, acceleration is in the high 70s. While not as high with respect to the binary task, acceleration appears to provide important incite when combined with other modalities. The top nine modality combinations for the binary task and the top 14 combinations for the 2-class categorical all have acceleration as a modality within their set of combinations. Figure 8.7 presents a comparison of modality combinations without acceleration for the binary and 2-class categorical tasks. However, it is important to note that there is variance among training runs enabling a modality combination to move slightly up or down the average performance scale; this is evidenced by the variation between the binary task and two-class categorical, Figure 8.7. Our conclusions here are more mixed. While having more modalities does tend to increase performance, it is not always the case that more is better. Even so, when all modalities are included the performance of the model is typically strong. After acceleration, there is not a clear modality that is always a top performer, but EDA tends to be higher performing. Pupillary response does seem to be selected when paired with other modalities, but is not favored strongly over measures from the E4 like PPG and temperature.

8.4.6. Performance Given Subject Experience

To understand how performance is affected by subject experience we built three RTLX family of models trained on only one subject type, each. That is we built three groups of models (33 models each) where each group was trained on one subject type: novices, operators, or pilots. Figure 8.8 depicts the accuracy of the 33 models for each group, which are defined by subject type, and were folded with cross-subject, 2-fold cross-validation, and
Figure 8.7: Equal weight RTLX family of models, ordered by average fold accuracy, all subjects whereby only two subjects were left out. This was done because of the dataset size constraints — 10 novices and 9 operators, which provided four and three folds, respectively — nine folds for the Pilot group. We again aggregated the folds of all models within each group. And we aggregated the accuracy across all modalities within a group and compared groups with a two-sided Wilcoxon signed-rank test [276] — (pilots vs operators: p-value=6.49e-05, pilots vs novices: p-value=7.1e-07, novices vs operators: p-value=4.94e-06). The Wilcoxon signed rank test is a non-parametric version of the paired T-test, and we used this test because the data was not found to be modeled well by a normal distribution. Both operators and pilots groups failed the Shapiro-Wilkes test [232] for normality. We thus concluded that the differences among each subject-type group are significant, and subject type can influence
the quality of the model. Moreover, the pilot group tends to have the strongest performance across all models, followed by operators, and then novices. For operationalizing this model as an instructional tool, this result is encouraging. We can expect more reliable performance for more experienced groups, and performance tends to be more reliable as the learners gain additional experience.

Figure 8.8: Equal weight RTLX family of models, ordered by average fold accuracy and trained only on: (a) Novices, (b) Operators, (c) Pilots, and (d) All subjects
8.4.7. Discussion

The implications of these results are mostly in the field of pilot workload awareness, but there are implications in the greater cognitive computing fields. Clearly, activity, maneuver (the human task), and skill level are correlated as indicated by our results that multi-domain classification is possible (though did not provide a distinct advantage); and they have an effect on model performance for classification of cognitive load. However, it is unclear how these three factors are inter-related based on our analyses. Gray [99] may have had the correct approach with his introduction of pilot inceptor workload. Inceptor workload may support a method of label alignment, as discussed in Section 8.2.1. Specifically, it may be possible to use the aggressiveness vs duty cycles method to label an observed maneuver without subjective ratings. As such, further investigation into activity and cognitive load is warranted.

The implication of modality choice is relatively clear. In our analyses the modality combination was a hyper parameter with some combinations outperforming others. While the use of many modalities was typically a good performer, it was never consistently better than using a subset of modalities, which has implications for a number of applications. Many applications can get a “good-enough” classification using only the sensors from the E4 wrist-band. Others may find a more reliable result with the inclusion of pupillary response. However other considerations need to be further investigated. These include investigation of modality fault tolerance, modality inter-change through context-aware computing, and modality latent data augmentation.

Additionally, it may be possible to reverse engineer the model to understand what aspects of a given modality are important for classification of cognitive load or other human cognition metrics. The exploration of activity with respect to workload must be understood.

By understanding the internal context of an aviator during a simulation, essential aspects of the learning experience could be efficiently exposed to a computing system. It is likely possible to personalize the learning experience for students, expediting the educational effectiveness by maximally challenging the learner without overloading them as hypothesized by
cognitive load theory [179]. Given our research showing that two levels of load are relatively reliable to infer, it should be possible to reformulate the theory of cognitive load (in learning) with this constraint. The utility of a binary cognitive load classifier may be enough for a number of personalized learning algorithms to come to fruition. The limitations of this are currently not well understood and require further research.

8.5. Edwards Pilot Workload Study

The Edwards Objective Measures of Pilot Workload study, discussed in Chapter 6, was a feasibility study to inform on utility of physiological sensor systems as a flight test data source for objectively assessing pilot workload. The SMU AT&T Center for Virtualization was one of three institutions that participated with our Empatica E4 capture software, participating in two of the sets of flight experiments. The study was conducted in real flight on a military cargo aircraft at Edwards AFB, where we evaluated the Empatica E4 wristband. This additional data collection scenario allowed us to ask: Do the models of cognitive load trained in our VR experiments generalize well to new pilots flying in real aircraft, under maneuvers not included in our original experiments?

8.5.1. Results

Ultimately, 33 maneuvers of workload data were collected averaging 7 minutes long, with a median of 6 minutes and a standard deviation of 4.7 minutes. Six maneuvers lasted over 10 minutes long, with one spanning 22 minutes. Due to the nature of the tasks, this data is unbalanced with only five samples of low load and 28 samples of high load. The unmodified subjective RTLX ratings were aligned and mapped to binary labels, as in Section 8.2.1. The modality data were processed and standardized as in Section 8.2.5. This new dataset was run-through the variable-Length $BM^3TX$ model with only the E4 modalities (PPG, acceleration, EDA, and peripheral skin temperature) with respective engineered features. The results are listed in Table 8.7. While the results are promising — an accuracy of 90.9% across 33 data points with five previously unobserved subjects in actual flight — the dataset is
too unbalanced for conclusive results. And in some cases, Pilot C and D, there is not enough data to make a conclusive judgment. Even so, a description of performance is warranted as an exploratory exercise. A hypothesis might be made that simulator data cannot capture the level of high load induced in real flight, and this dataset is skewed for an observance of high load with respect to the model trained on simulator data. However, Pilot C’s mis-classified sample is a low-load label classified as high-load, while Pilot D and E, each, classified a high load sample as low load — one type I error and two type II errors. Ultimately, we can only conclude that this methodology is feasible. More flight data is needed for a conclusive result, but the results are quite promising.

Table 8.7: Variable-length $BM^{3}TX$: Metrics from the classification of Edwards Pilot Workload Study flight data — five test pilots across two flights flown August 28th and August 30th, 2018.

<table>
<thead>
<tr>
<th>Subject(s)</th>
<th>Count</th>
<th>Binary</th>
<th>Acc.</th>
<th>TPR</th>
<th>Prec.</th>
<th>AUC</th>
<th>2-Class</th>
<th>Acc.</th>
<th>TPR</th>
<th>Prec.</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>All Pilots</td>
<td>33</td>
<td></td>
<td>90.9%</td>
<td>90.9%</td>
<td>90.4%</td>
<td>.936</td>
<td>93.9%</td>
<td>93.9%</td>
<td>93.9%</td>
<td>.957</td>
<td></td>
</tr>
<tr>
<td>Pilot A</td>
<td>14</td>
<td></td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>1.0</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Pilot B</td>
<td>6</td>
<td></td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>1.0</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Pilot C</td>
<td>2</td>
<td></td>
<td>50.0%</td>
<td>50.0%</td>
<td>25.0%</td>
<td>1.0</td>
<td>50.0%</td>
<td>50.0%</td>
<td>25.0%</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Pilot D</td>
<td>3</td>
<td></td>
<td>66.67%</td>
<td>66.7%</td>
<td>44.4%</td>
<td>1.0</td>
<td>100.0%</td>
<td>100.0%</td>
<td>100.0%</td>
<td>1.0</td>
<td></td>
</tr>
<tr>
<td>Pilot E</td>
<td>8</td>
<td></td>
<td>87.5%</td>
<td>87.5%</td>
<td>93.75%</td>
<td>.857</td>
<td>87.5%</td>
<td>87.5%</td>
<td>93.75%</td>
<td>.857</td>
<td></td>
</tr>
</tbody>
</table>

8.6. Conclusions

In this research we use the aviation domain for inducing and evaluating cognitive workload. We evaluated numerous machine learning models to classify cognitive load across subject and across task. We find that we can accurately and reliably use multiple physiological/biometric modalities to objectively evaluate cognitive load in a simulated flight environment.
The activities of this research involved the collection of physiological measures relevant to aviators as they underwent simulated flight scenarios for both low and high levels of induced workload. We proposed a new concept of subjective label alignment as a way of reducing the noise inherent in subjective cognitive load ratings. We highlighted that classification given the mental load characteristic is the wrong path towards objective measure of workload across subjects. We further proposed two versions of a new method for cognitive load classification: one with fixed-width input, and the other variable-width input. These architectures took advantage of X-Vector like architectures and deep, multi-modal, multi-task, and generative learning, with many 1-D modalities from physiological sensors encoded into a latent space. Statistics layers are calculated and passed through to the bottleneck, and finally passed through to multiple task layers classifying multiple tasks across 40 subjects inclusive of three subject types — pilots, operators, and novices. Our approach was validated with real-flight data from five military test pilots collected over two test and evaluation flights on a C-17 aircraft.

The importance of measuring human performance objectively across subjects is paramount, and the results of this research suggests cognitive load can be objectively classified across subject and activities into at least two levels with reliable accuracy and in near-real time, > 81.0%. Additional levels of sensitivity should and need to be investigated further. Interestingly, the skill level at the chosen tasks influence the accuracy of the model, with more reliable results for pilots with increased flight experience. Furthermore, this methodology for classification of cognitive load may be of use in establishing the context of an aviator while in-flight. The scope of our conclusions is limited to scenarios from our experiments, but the classification of cognitive load across maneuvers for additional flight tasks or for other activities in other domains are likely applicable. We leave the investigation of cognitive load classification in additional flight activities and other disciplines to future work.
Chapter 9

THE CLASSIFICATION OF GAZE SCAN PATTERNS

9.1. Introduction and Motivation

Using ubiquitous computing for augmenting learning environments is a key application area highlighted in the UbiComp community. In her re-framing of the vision of UbiComp, Yvonne Rogers described the classroom as an under-explored area in which UbiComp technologies might have far reaching impact [205]. She argued, the ambience in classrooms, while varied, is more limited than the world in general, making it possible to design and evaluate UbiComp technology in a somewhat narrowed context—learning. Many research papers have explored the idea of sensing and understanding users in a learning environment such as (among many examples): using eye-tracking for assessing students performing mental math [192], using electroencephalography for assessing if a student was in a state of flow [268], and using brain-computer interfaces for assessing piano students [281]. In this chapter, we evaluate how using eye-tracking (with machine learning) can be used in a specific educational environment where wearable technology is already ubiquitous—training aviators to fly aircraft maneuvers in virtual reality when being observed by a flight instructor (a scenario typical for training military personnel to fly specialized aircraft). One area of key interest is to reduce the need for direct instructor observation of an aviator, especially for monitoring the gaze/scan patterns of pilots. When evaluating the performance of aviators, flight instructors rely on observation and after-the-fact assessment. Scan patterns are vital aspects of context for a flight instructor, and are fundamental to basic flight. For example, a student might scan too rapidly, omit, or fixate — these are common errors when scanning the horizon and cross-checking instruments [259]. Anecdotally, flight instructors often cite that head and eye movements are pivotal for judging student intent and situational awareness.
Virtual reality-based training environments with embedded eye-tracking offer the possibility to automate and provide more context to some aspects of instructor observations and potentially expedite the learning process. Therefore, a key research question is: Can machine learning accurately evaluate gaze pattern quality for aviators in various phases of flight?

A gaze or scan pattern is a technique in which an aviator observes all requisite information inside and outside the aircraft in-order appropriately and safely fly that aircraft. The scan begins and ends in the same position, observing all applicable items – “systematically, thoroughly...complete, and continuous” [262].

We propose a new method for gaze classification by transforming gaze or scan patterns into heatmaps and classifying them with deep convolutional neural networks [139]. The patterns are classified into levels of “quality” that would typically require review from an instructor. Data are collected in a mixed-reality training environment using a physical flight simulator, a virtual reality environment, and a gaze-tracking sensor for monitoring eye movements within the virtual space. From these devices a heatmap is synthesized from the pattern created by the gaze of an aviator flying during a specified window of time. We detail the contributions of our work as follows:

- We designed and carried out a human subjects experiment for aviators in a variety of flight scenarios, discussed in Section 9.2.5. We recruited three instructors (Section 9.2.5) to review the gaze patterns from these scenarios (discussed in Section 9.3), and we analyzed inter-rater reliability; and we concluded there is strong agreement among these expert raters on what gaze quality is for a given maneuver.

- We proposed methods for gaze data augmentation specific to pilot scan patterns that increase robustness of trained machine learning models (Section 9.5)

- We investigated two competing, convolutional neural network architectures: a task agnostic model and a multi-task model (Section 9.4); and we evaluate the architectures with K-fold cross-validation, achieving greater than 93.0% average test accuracy compared to instructor observation (Section 9.6).
The location of gaze within a reference frame, i.e., what the aviator is looking at as mapped to that reference frame, is based on: (1) the alignment of head and eye positions with respect to the subject’s field of view (FOV), (2) the object observed, (3) the location on the object observed, (4) the reference frame into which the point-of-gaze is mapped, (5) the angular error and precision of that translation, and (6) the calibration error — therefore the mapped position within that reference frame will vary. Thus, the problem is not as simple as pinpointing the objects within the reference frame and defining a bounding box because the regions of interest for a pilot scan pattern will vary depending on the maneuver in question, and the number of regions within that reference frame can increase based on the complexity of the maneuver. A classifier must both scale with increased regions of interest and handle perturbations in gaze position as projected onto the reference frame. This is not just for fixed head position (among aviators), but in dynamic support of the yaw, pitch, and roll of head and eye movements of a given aviator — especially as it pertains to the movement of the subject’s FOV about the heads-up display’s (HUD) eyebox [242]. An obvious fixed-position example is that varying aviators have different abdomen heights and seat height preferences, which can lead to some aviators looking slightly downward or slightly upward towards the instrument panel and displays [261]. With convolutional neural networks (CNNs), the representations learned are robust to translations [46]. CNNs are capable of handling fixed and dynamic perturbations throughout a reference frame, and do not require individual labeling of regions of interest. Given the advantages of CNN, we hypothesize these models are superior in classifying gaze.
9.2. Background and Related Work

Our work builds from a number of research communities. As such, we divide our discussion into four relatively disjoint areas: (1) sight picture, (2) heatmaps in eye-tracking, (3) gaze in aviation, and (4) other works in gaze classification.

9.2.1. Sight Picture

In the case of firearms “sight picture” can be referred to as the perspective, as viewed by the shooter, created by “the alignment of the sights of a firearm with the target” [159]. The concept of “sight picture” in aviation originated with fixed gunnery weapons [164]. When such a firearm is affixed to an aircraft the whole aircraft must be maneuvered in order to properly aim. Thus, an early concept of sight picture in aviation can be thought of as the perspective of the pilot—what he/she sees through a reflector plate of the cockpit—given the aircraft serves as the firearm. This requires perceptual abilities to evaluate, aim, and fly the aircraft [164]. Students would learn a series of sight pictures for a discrete number of angles of attack (AoA), which is further compounded by the realities of a moving target in air-to-air situations. Strong “perceptual memory” is required for both understanding what the correct sight picture is and correcting to it, given the current sight picture [164]. Applications and technology have evolved, but the concept of aligning an aircraft with the environment for the purpose of accomplishing a specific maneuver based on an understood mental picture of what is correct—from the perspective of the pilot in the seat of the cockpit—has become fundamental to basic flight [83, 133, 260–262].

9.2.2. Heatmaps in Eye-Tracking

Gaze can be defined as the direction of the visual axis within a reference frame. It is a summation of eye position relative to the head, and the head position relative to the same reference frame [102]. The visualization of gaze is a key research area in gaze tracking. A predominant gaze visualization is the heatmap, which can provide clear depiction of aggregate gaze by combining gaze fixations while sacrificing the depiction of the order in which the
Figure 9.1: Aggregate high-resolution heatmap of a normal landing

fixations occurred [71]. Privitera [193] found that different subjects are mostly consistent on what regions they observed, but are less consistent in the order they view them. Therefore, order is not necessarily as important as the locations observed [71]. Špakov provided an in-depth examination on the methods for visualizing fixations, including heatmaps [283, 284]. Newn et al. [171] showed, using nine gaze representations, that humans have a strong capacity to accurately infer intent. Stellmach et al. [247] examined gaze fixation in three-dimensional (3D) virtual environments surveying 3D scan paths, 3D heatmaps; and they provide a prototype toolkit for aiding future eye-tracking studies.

Given heatmaps are accumulated fixations, Stellmach et al. found that they are useful for indicating visual attention over a period of time because visualizing data in three dimensions enables a representation over a longer period of time. We chose to use three two-dimensional heatmaps within fixed reference frames around aviators because: (1) experienced aviators have highly developed perceptual memory and judgement [164], (2) heatmaps provide a clear depiction of aggregate gaze [71], (3) subjects are consistent on regions they observed [193], (4) gaze order is not necessarily as important [71], (5) heatmaps are useful for indicating visual attention over a period of time [247], and (6) humans have a strong capacity to infer intent through gaze representation [171].
In our experiments, we segmented flights into smaller phases, where similar gaze region abstractions can be expected. The three reference frames provide full cockpit coverage. As an example, Figure 9.1, is a high-resolution heatmap aggregated over a full normal landing maneuver (the clouds are part of a background image only, and do not reflect what the pilot actually saw during the landing). The three reference frames are shown in the Figure and overlaid with heatmaps denoting where the pilot scanned, from the perspective of each frame. From the heatmaps, it is apparent that the pilot looked directly at the HUD most of the time, while periodically scanning the horizon.

9.2.3. Gaze in Aviation

Our work is not the first to look at eye tracking with aviation. However, previous works do not attempt to classify the quality of a gaze pattern. Weibel et al. [274] looked at digital ethnography to understand visual attention of aircrew throughout varying phases of flight using a mobile eye-tracking system — they reported on techniques and methods to digest and visualize the dynamics of time-synchronized, multimodal, visual attention data. Specifically, they looked at visualizing tracking data, analyzing areas-of-interest, with infrared markers and the errors associated, visualizing the temporal dynamics, such as overlaying gaze on video frames, and gaze-to-object recognition. Weibel et al. sought to discern when an aviator’s gaze fixed on an object of interest without IR markers. They did this with OpenCL by matching objects from one frame to all frames. Vrzakova and Bednarik [265] sought to understand how mobile eye-tracking could work in a real cockpit. Recently, lounis et al. [150] looked to enhance aircrew-aircraft interaction. They monitored the attentional behavior of aircrew using a gaze tracker and developed a cockpit monitoring database that serves as an assessment tool. They expanded on their work by developing a flight eye-tracking assistant built on their database that uses thresholding of dwell times for areas of interest with audible alarms [151]. While these previous works investigated physiology in the context of aviators, none explicitly address quality of gaze as observed by a subject matter expert.
9.2.4. Other Works in Gaze Classification

A number of works have investigated the classification of intent and attention from gaze data. While looking at a way to discern intent, Goldberg and Schryver [93] developed heuristics from multiple discriminant analysis to enable gaze controlled UI zoom. Frutos-Pascual and Garcia-Zapirain [88] looked at attention performance with saccadic and fixation gaze data over 32 children. They achieved 88.0% accuracy with a random forest classifier. Abdelrahman et al. [3] developed a way of classifying attention types through the use of thermal imaging and eye tracking. They developed several classifiers capable of classifying four types of attention [241] with an average AUC of 80.3%. Similar research to the work Weibel et al. conducted, Barz et al. [19] examined using gaze-to-object recognition with neural networks to classify objects and ensure the user draws attention to that object. They used a dispersion algorithm for fixation [18] and thresholds for attention. Interestingly, a significant step towards gaze classification was conducted by Li et al. [145]. They investigated classifying gaze itself by breaking it into three fundamental patterns: (1) saccades [39, 102], (2) smooth pursuits [202], and (3) fixations [201]. They studied consumer helicopter drone pilots, while performing guidance and control tasks, as well as surgeons who conducted a peg transfer task [246]. They devised a scheme for converting gaze to a fixed reference frame for classification. They employed a spherical head centric coordinate frame, from a study of the receptive field of flies [120], correlated with six-cameras. Using both empirical thresholding and hidden Markov models (HMM) to classify gaze data, they were able to accurately classify the three fundamental patterns with the use of gaze velocity and distance. Our work is similar to these in that we classify scan patterns based on discrete criteria.
Our work is similar to these in that we classify gaze patterns based on discrete criteria. However, we use multiple reference frames, a convolutional neural network architecture, and a more complex classification task — equating scan patterns to human observation. Because the complexity of the classification task is increased, we also investigate multiple reference frames and a convolutional neural network architecture with increased predictive power. The advantage of this methodology is its ability to handle both perturbations in gaze position and scale with more complex gaze regions as the task and gaze patterns increase in complexity—all without the use of bounding boxes.

9.2.5. Expert Review

Once data collection was completed, the screen capture video was separated by subject and maneuver. Heatmaps were generated, using a non-overlapping 30-second sliding window for the duration of each maneuver flown. Three subject matter experts labeled a subset of the gaze data using maneuver videos and heatmaps to establish inter-rater reliability. The subject matter experts included two experienced instructor pilots (IP) and one experienced instructor combat systems officer (CSO). Further details on the review process are discussed in Section 9.3. Table 7.1 lists the subject demographic information over 40 test subjects. This includes the sample mean, standard deviation, min and max values for subject age, flight hours, and number of aircraft flown. Further, the percentages for type of flight experience are listed. Table 9.1 lists the demographic information for each rater.

<table>
<thead>
<tr>
<th>Instructor Pilot I</th>
<th>Instructor Pilot II</th>
<th>Instructor CSO*</th>
</tr>
</thead>
<tbody>
<tr>
<td>27 years flying</td>
<td>40 years flying</td>
<td>13 years flying</td>
</tr>
<tr>
<td>12 aircraft flown</td>
<td>16 aircraft flown</td>
<td>31 aircraft flown</td>
</tr>
<tr>
<td>5,025 total flight hours</td>
<td>12,860 total flight hours</td>
<td>1,300 total flight hours</td>
</tr>
<tr>
<td>2,000 instructor hours</td>
<td>1,250 instructor hours</td>
<td>225 CSO instructor hours</td>
</tr>
<tr>
<td>USN TOPGUN graduate and instructor</td>
<td>USN TOPGUN graduate and instructor</td>
<td>USAF Test Pilot School graduate</td>
</tr>
</tbody>
</table>

*Total hours encompass both CSO and pilot time
9.3. Gaze Quality Labeling

“Gaze quality” in the context of this research is simply a rating on a 3-class ordinal scale evaluating the subject’s ability to scan his environment in-order to safely and correctly execute the assigned maneuver or task. It was based on instructor opinion among the three raters. In order to establish an appropriate scale, inter-rater reliability (IRR) was evaluated. The raters included two IPs and an instructor CSO. All three are seasoned instructors and evaluators with military experience. Given their experience, Section 7.3.1, they have refined perceptual abilities, Section 9.2.1, making them ideal for the labeling task. Once IRR was established, the CSO labeled the remaining dataset for both single- and multi-task models.

Figure 9.2: Example of a concatenated video frame from a video that instructors reviewed for labeling gaze quality. Three heatmaps of different screenspaces line the top of the frame and the bottom consists of video from the pilot (left) and a zoomed heatmap of the HUD (right).
In support of our labeling effort we created high-resolution (hi-res) heatmaps for the instructors to review along with the maneuver video. These hi-res heatmaps, used in conjunction with maneuver video, are useful in labeling the quality of segments of gaze data over a window of time instead of a specific heatmap and resolution.

9.3.1. Review Interface Creation

The data collection videos and data were segmented based on subject and maneuver. Hi-res heatmaps were generated over 30-second, consecutive, non-overlapping windows during a maneuver. If there were any leftover data points, a final overlapping 30-second heatmap was generated for the last 30-seconds of the maneuver. The start and stop times of these 30-seconds windows were saved so that our machine learning model could process the data in the same labeled time window. A 30-second window was chosen because the raters considered that interval to be a reasonable amount of time for an instructor to observe and discern the quality of a student’s scan pattern.

The hi-res heatmaps were created by using a bivariate kernel density estimate (KDE) with a Gaussian kernel. A 500-level KDE overlay was generated on top of still images of the cockpit from each screenspace’s FOV, as shown in Figure 9.2 (Top). The heatmap intensity was scaled over all three screenspaces. A zoomed version over the HUD was also created.

The interface used by the instructors for labeling consisted of the three screenspace heatmaps and a zoomed HUD heatmap concatenated to each video frame of the pilot’s maneuver. This means that as the reviewer was observing the maneuver video, he was also viewing the hi-res heatmaps for the 30-second window of data being observed. The original maneuver video provided SA on aircraft movement and position, but it also provided gaze convergence tracking represented by a green eye floating about the cockpit. This way, the instructors could review not only the heatmaps synthesized from gaze data, but they also observed what the pilot was staring at during their maneuvers. A reviewer always reviewed data for an entire maneuver and the windows of data were labeled in the order they appeared in the maneuver. Figure 9.2 provides an example of a concatenated video frame.
9.3.2. Rating Process

For labeling, the three raters used a grading scheme of poor, fair, or correct. Within this scale the raters were further allowed to rate windowed data with scores that were “in-between” levels such as “poor-to-fair,” yielding a five-class scale: (1) poor, (1.5) poor-to-fair, (2) fair, (2.5) fair-to-correct, and (3) correct. Pilot I reviewed 520 sets and Pilot II reviewed 517 sets of thirty-second windowed data, pulled from across all 40 test subjects. The information reviewed included the convergence tracking video with the concatenated heatmaps discussed in Section 9.3.1.

Both pilot raters reviewed 109 gaze patterns that overlapped between the two datasets. This was to support the investigation of IRR between the two pilot raters. The CSO then reviewed both datasets, as he was the primary annotator of the full dataset. The label count distributions are found on Figure 9.3.

9.3.3. Inter-Rater Reliability Results

When investigating IRR, we employed Cohen’s $\kappa$ [49, 50], the correlation coefficient for the binary-rater case, Fleiss’ $\kappa$ [84] and Randolph’s $\kappa$ [199] for the multi-rater case. We investigated multiple levels of agreement by transforming the 5-class ratings into multiple 3-class variations. Specifically, the 5-class was transformed into 3-class ratings by ceiling and flooring the class labels. A final 3-class version was calculated by taking the floor of the rating 1.5 and the ceiling of the rating 2.5. The results of each transformation and evaluation criteria are shown in Table 9.2.

For the binary-rater case, pilots I and II’s Cohen’s $\kappa$ were strongest when the floor and ceiling were utilized, a $\kappa$ of 0.71. Further, an r-value of 0.79 reveals a strong positive linear relationship. Pilot I and the CSO had less agreement than that of both instructor pilots. However, when the ceiling method was used a $\kappa$ of .4 reflects moderate agreement. This was further observed by an r-value of 0.46 — indicating a positive linear relationship. Finally, Pilot II and the CSO exhibit strong agreement with a $\kappa$ of 0.71 and an r-value of 0.76 when the floor and ceiling method was utilized. Overall, this signifies that Pilot I and the CSO
have moderate agreement while both have strong agreement with Pilot II.

The multi-rater case yielded a strong inter-rater relationship. Again, the ceiling and floor/ceiling methods provided the highest reliability. The overall multi-way Randolph’s $\kappa$ was 0.78 and the multi-way Fleiss’ $\kappa$ was 0.56, indicating a strong inter-rater reliability was established.

Of note, the rating label distribution is uneven and further addressed in Section 9.5. While both the ceiling and the floor/ceiling methods provide high inter-rater reliability, the floor/ceiling method was chosen as it yielded more labels for ”poor” class — the weakest class count across tasks from the unaugmented dataset.
### Table 9.2: Inter-rater Reliability

<table>
<thead>
<tr>
<th>Raters</th>
<th>Coef.</th>
<th>5–Class</th>
<th>Ceil [3–Class]</th>
<th>Floor [3–Class]</th>
<th>Floor/Ceil [3–Class]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot I and II</td>
<td>Cohen’s κ</td>
<td>.45</td>
<td>.60</td>
<td>.36</td>
<td>.71</td>
</tr>
<tr>
<td></td>
<td>r</td>
<td>.66</td>
<td>.68</td>
<td>.50</td>
<td>.79</td>
</tr>
<tr>
<td>Pilot I and CSO</td>
<td>Cohen’s κ</td>
<td>.30</td>
<td>.40</td>
<td>.24</td>
<td>.37</td>
</tr>
<tr>
<td></td>
<td>r</td>
<td>.43</td>
<td>.46</td>
<td>.33</td>
<td>.44</td>
</tr>
<tr>
<td>Pilot II and CSO</td>
<td>Cohen’s κ</td>
<td>.63</td>
<td>.66</td>
<td>.60</td>
<td>.71</td>
</tr>
<tr>
<td></td>
<td>r</td>
<td>.78</td>
<td>.72</td>
<td>.68</td>
<td>.76</td>
</tr>
<tr>
<td>Pilot I, Pilot II, and CSO</td>
<td>Fleiss’ κ</td>
<td>.30</td>
<td>.55</td>
<td>.39</td>
<td>.56</td>
</tr>
<tr>
<td></td>
<td>Randolph’s κ</td>
<td>.47</td>
<td>.78</td>
<td>.50</td>
<td>.78</td>
</tr>
</tbody>
</table>

Figure 9.4: Four example center screenspaces (a-d) — While (d) would rate “poor” for all 3-tasks, (a-c) are each characterized as (a) “correct”, (b) “fair”, and (c) “poor” given the climb, cruise, and decent task. However, (b) is labeled “correct” for the ground task, as only airspeed and centerline are needed, and (c) is labeled “correct” for the final approach task, a pilot flying the AoA bracket.

#### 9.3.4. Multi-task Labeling

Throughout the labeling of the dataset, raters expressed concern that there can be more than one phase within a maneuver. For example, on takeoff the pilot transitions from ground roll to a climb. Each of these phases can have a different “correct” gaze pattern. There is a potential robustness issue for a machine learning model because a given gaze pattern might be judged “correct” in one phase, but would garner only a “fair” in another phase, see Figure 9.4. A more robust solution is to use a multi-task machine learning model that has the ability to interpret the same gaze pattern differently, depending upon the phase. From the maneuvers we grouped the phases into three generalized phases (tasks) according to
the relative similarity of gaze pattern for the phases: (1) climb, cruise, and decent (CCD), (2) ground, and (3) final approach. Taking advantage of the idea that there are common gaze patterns among phases, the annotator was asked to label every window of gaze data according to all three generalized phases. That is, the annotator labeled quality as if the subject was flying each of the three generic phases described above, and providing a label for each — Figure 9.3.

### 9.4. Deep Learning Architecture

For our convolutional architecture we employed two key techniques: transfer learning and multi-task learning. Leveraging prior knowledge to hasten the learning of new tasks is known as transfer learning [128, 183]. These methods preserve and take advantage of previously trained models from one task or domain and apply them to a second different task or target domain. More broadly, transfer learning allows for domains, tasks, and distributions between training and test to be different [183]. Such methods can effect new training of accurate models for an entirely different task and/or source domain where labeled data may be limited [183].

A related concept is the preservation of learned knowledge while training multiple tasks — multi-task learning. While multi-task learning can be considered a form of transfer learning, it traditionally differs in that the shared knowledge is learned at the same time, between tasks, and during the training process. A typical approach for multi-task learning is to uncover the shared latent features that can benefit each task [183]. Ruder, [210], showed that multi-task learning models tend to prefer solutions that generalize.

In this work we took advantage of the learned weights from the Visual Geometry Group’s 16-layer model (VGG16) from [234]. VGG16 was trained on an ImageNet repository ILSVRC-2012 dataset [211], a repository used for the 2012-2014 Large Scale Visual Recognition Challenge (ILSVRC). The ILSVRC-2012 dataset was built as a subset of ImageNet’s [58] greater repository with a training dataset of 1000 categories and 1.2 million images. We sought to take advantage of the spatial representations learned at the more shallow depths of the
VGG16 model—where such representations are less complex, domain specific, and more applicable to the gaze domain.

Two models were implemented — a task agnostic version, Figure 9.5, and a multi-task version with three tasks, Figure 9.6. Both models utilize the first seven weight layers of the VGG16 pre-trained model (pruned at the third max pooling layer), with all layers frozen during training. The input remains a 224 x 244 3-channel tensor as with the VGG16 model. However, the three channels are no longer red, green, and blue. Instead, each channel consists of a screenspace — left, center, and right — from the heatmaps synthesized by the test
subject’s gaze pattern, discussed further in Section 9.4.1. Because this implementation utilized the Tensorflow version of VGG16, all input tensors were standardized between -1 and 1. The pruned VGG16 model’s output is routed through two separable convolution layers [47] with 256 filters each, a kernel size of 3x3, and same size output padding. That output is then passed through a max pooling layer, with 2x2 pooling window size, and flattened. The flattened output is passed to three dense layers with a 0.5 dropout rate [245]; each subsequent layer is a step down by a power of two of the previous dense layer. Specifically, 128, 64, and 32 nodes, form the bottleneck. Finally, for the task agnostic model, one 32-node dense layer is followed by a three-node dense layer with a softmax activation [24,96]. For the three-task model, the bottleneck output is routed to three individual 32-node dense layers. Each are followed by a three-node dense layer with softmax activations. All hidden layers utilize a ReLU activation function [105, 126, 168]. We used a batch size of 64, but did not implement batch-normalization [122], given the suggestions of Simonyan and Zisserman [234].

![Figure 9.7: Center Screenspace Scaled 25 X 25 (A) Count Matrix, (B) Heatmap, (C) Gaussian Blur, and (D) Actual 244 x 244](image)

9.4.1. Input Heatmap Generation

For input heatmap generation the gaze data provided by the BBXR was mapped to a two-dimensional tensor. The model input is of the shape 244 x 244 with three channels, the same shape as with VGG16. However, rather than using the RGB channels, each channel is a distinct screenspace generated from the gaze data over a thirty-second window, left,
center, and right — in this order. While this method may introduce errors in the input distribution of VGG16, we did not observe any worrying behavior of the model; the VGG16 layers remained frozen and were not fine-tuned. A KDE was not used to create the heatmap input tensor, as in Section 9.3.1. Instead, to create the desired tensor we first mapped a consecutive thirty seconds of data to the desired resolution, 244 x 244. This was done by multiplying all values $\geq 0$ and $< 1$ by 244 for both x and y values. Second, a count matrix was generated, where a 244 x 244 matrix of 0s is created and a 1 added to the position of each x and y pair. Third, the maximum value was calculated across all three screenspaces because the heat or intensity was measured over all channels in the tensor, and all non-zero values are divided by this maximum value. Fourth, for a smoother input, a Gaussian blur with std of 1.0 was taken over the newly generated heatmap. Ultimately, this creates a lower resolution version of the Hi-def heatmap, for a given window of data, used for labeling. A scaled example of the center screenspace as it is transformed to a heatmap is shown in Figure 9.7. Finally, all three screenspaces were stacked creating a 244 x 244 x 3 tensor.

9.4.2. Training

For training we use the ADAM stochastic optimization method [135] with a learning rate range of 1e-6 to 1e-3, standard beta values, and gradient clipping range of 0.0 to 0.5. We chose the best hyper parameters based on the cross-validation results. Before training, we randomly separated 10\% of the pilots for a final test set. The remaining 90\% of the data were used for a 10-fold, across-subject cross-validation. That is, a given aviator cannot simultaneously be in a training fold and a validation fold. This resulted in validating folds consisting of about three aviators. We load balanced (i.e., stratified) these test folds such that at least one of the three subjects was always a novice or operator and one was always a pilot. Otherwise, the three subjects were chosen at random. Each fold was trained for fifty epochs. A final model was then trained with all the data, except the 10\% portioned test dataset.
9.5. Data Augmentation

For deep learning applications, it is ideal to have as much data as possible for training. In the absence of numerous labeled data, an augmentation process can help to synthetically boost the number of training samples. This is commonly known as data augmentation — using existing labels and manipulating the input data to create new samples. Augmentation is only used for the training samples. That is, the testing samples remained unchanged.

For augmenting gaze data, we implemented several augmentation methods. These included: (1) removing or clipping a portion of the heatmap, (2) perturbing the heatmap within the reference frame along an axis, (3) mirroring the heatmap across the vertical or horizontal axes, and (4) having the rater label gaze patterns from unassociated maneuvers according to proper sight picture. This process resulted in approximately 9,400 training samples across all gaze quality classes in addition to the original 3,877 samples.

The clipping method takes advantage of the sight picture awareness and further modifies a window of data by creating a new heatmap. This involves generating a heatmap from a portion of the windowed data and relabeling the new pattern respective to the given phase. For this research we only modified heatmaps created from “correct” labeled data. For example, we looked at the CCD generic phase, and aggregate the windowed data that are labeled “correct.” These gaze patterns tend to have a triangle-like shape over the HUD about the airspeed, altitude, and pitch ladder/flight path marker (FPM). We created a new heatmap by removing the left portion, any data with less than a ratio of 0.49 for the x-coordinate — gaze convergence over the airspeed indicator. We repeated this process again creating a second new heatmap by clipping the right side, any data greater than a ratio of 0.51 for the x-coordinate — gaze convergence over the altitude indicator. This method added twice the size of CCD class “correct” labels to the class “acceptable” labels. Post clipping, these heatmaps were verified or relabeled manually by the annotator and normalized by their largest value.
For this research we only perturbed the heatmaps vertically along the y-axis. Specifically, the heatmaps were adjusted up or down by ratios between -0.03 and 0.1. The heatmaps retained their original labels, and were further verified by the annotator. This method doubled the available training samples for our convolutional networks.

The mirroring method used flipped the heatmap about an axis, and then labeled appropriately given the sight picture for each task. We only utilized mirroring about the y-axis. This was particularly useful for “poor” labeled gaze data that has heavy gaze fixation on one side or the other.

Finally, gaze heatmaps from the other maneuvers, Section 9.2.5, were labeled. In conjunction with the concatenated videos we took advantage of the perceptual awareness of the raters, Section 9.2.1, by having the annotator mentally project the appropriate phase sight picture onto the maneuver being flown — both BAT and air intercept tasks. Augmented gaze pattern label distributions are found on Figure 9.3.

9.6. Results

As discussed in Section 9.4.2, the training of each model included the 10-Fold, cross-subject, cross-validation followed by a final fit over the entire dataset minus the 10% set aside for the test dataset. Results for both the 10% test set and the averaged cross-validation are presented. The training for both the task agnostic and multi-task models converged on or about the 20th epoch. The results of the task agnostic and multi-task models are discussed below, followed by a second inter-rater reliability analysis comparing the two models to the human gaze quality raters.

9.6.1. Task Agnostic Model

Figure 9.8(a) presents the task agnostic model’s confusion matrix for both the average accuracy and average categorical true positive rates over all ten folds. Figure 9.8(b) characterizes the confusion matrix for test accuracy and true positive rates for each category.

The task agnostic model has an average fold accuracy of 89.9% with a test accuracy
of 89.2%. The average true positive rates over all folds is 93.0% for poor, 89.0% for fair, and 84.0% for correct. The test dataset true positive rates are slightly different for each quality label with 90.0%, 68.0%, and 93.0% for poor, fair, and correct. The fair case is likely reflective of the potential task agnostic issues discussed in Section 9.3.4. That is, an unseen heatmap from the test dataset is potentially classified incorrectly because it can have two different quality labels depending on the flight phase—and the agnostic model has no information regarding flight phase. Ultimately, this model accurately classifies gaze patterns that are poor 90.0%, fair 68.0%, correct 93.0% of the time. This may imply that a “correct” or “poor” gaze pattern generalizes across a number of flight phases, whereas fair patterns are more dependent on phase.

9.6.2. Multi-task Model

Figure 9.9 (a-c, top row) presents the multi-task model’s confusion matrices for both the average accuracy and average categorical true positive rates over all ten folds for each task. Figure 9.9 (d-f, bottom row) depicts the confusion matrix for test accuracy and true positive rates for each category. The “final approach” task was converted to a binary classification because too few examples were labeled as “fair” by the raters. Because there were not enough ground truth labels to train the model, we only reported the binary classification “poor” versus “correct.”
The multi-task model has an average fold accuracy of 94.2% with a 93.0% average test accuracy across all tasks. Individually, the average fold accuracy for each task is 89.1%, 90.2%, and 94.63% with test accuracies of 91.0%, 91.8%, and 96.45%, for CCD, ground, and final approach, respectively. The test dataset true positive rates are comparatively stronger for each task than with our task agnostic model. CCD classifies 91.0%, 86.0%, and 94.0%, while the ground task classifies 92.0%, 88.0%, and 94.0%. Finally, the final approach task classifies 96.0% and 100.0% for “poor” and “fair” each. These results imply that the model is robust to different tasks, but they do not indicate if the model is comparable to a human instructor. To elucidate this, we turn to measures of inter-rater reliability, including the model, as if it were a fourth instructor.
9.6.3. Human-Model Inter-Rater Reliability

We compared inter-rater reliability measures for the task agnostic model and multi-task model, separately. This analysis is identical to that carried out in Table 9.2 among raters, except we treated each model as if it were a human instructor. Table 9.3 presents the results of the IRR analysis between each rater and model, among all raters and each model, and among a subset of raters and each model. This analysis helps to answer the following question: Does the model agree with human raters in a manner that is similar to how human raters agree with each other?

The **task agnostic model** has Cohen’s $\kappa$ of .18, .37, and .43 with Pilot I, Pilot II, and the CSO, respectively. These $\kappa$ values signify some agreement with Pilot I and moderate agreement with Pilot II and the CSO. For the multi-rater variants, the Fleiss and Randolph $\kappa$’s are .39 and .61, establishing moderate IRR agreement according to the scale defined by Landis and Koch [143].

For the **multi-task model** strong inter-rater reliability is achieved. The multi-task model and Pilot II have Cohen’s $\kappa$ of .67, substantial agreement. The relationship between the CSO and the model has a $\kappa$ of .70, also a substantial agreement. This indicates strong
agreement, especially given that CSO and Pilot II have a Cohen’s $\kappa$ of .71 (Table 9.2). Furthermore, pilot II and the CSO each have an $r$ value with the multi-task model that signifies a strong linear relationship, with .73 and .75 respectively. The least agreement is between the multi-task model and Pilot I, which has a $\kappa$ of .23. While lower, this is similar to the CSO and Pilot I, which have a $\kappa$ of .37 (Table 9.2) — the lowest agreement among human raters.

For the multi-rater case, the model and all three raters have a Fleiss’ $\kappa$ of .57—which is a slightly better Fleiss’ $\kappa$ than that of the all human raters, .56 (Table 9.2). Both rater combinations have the same Randolph’s $\kappa$ of .77, signifying substantial agreement. If Pilot I is removed from the multi-rater analysis, the highest multi-rater $\kappa$ values are achieved, .66, Fleiss, and .85, Randolph — almost perfect agreement.

In summary, for IRR among the raters and the models, it is apparent that both models are similar to the respective relationships among the full dataset annotator, the CSO, and the two pilot raters — that is the CSO and Pilot I having shown less agreement than the CSO and Pilot II.

9.7. Heatmap Examples

The following are examples of poor, fair, and correct labeled heatmaps for each model task (flight phase) over a 30-second window. Note that this is not exhaustive, for example Figure 9.10 (c) is also “correct” for the final approach task.
For each example in Figure 9.13, these heatmaps are negative examples and are labeled poor for all three model tasks (flight phases) — CCD, ground, and final approach. All cover a 30-second window period.

9.8. Model Training Hardware and Software

For this research, we used the Python versions of Scikit-Learn, TensorFlow and Keras. Both video concatenation with high-resolution heatmaps and model training were conducted on the Southern Methodist University ManeFrame II high-performance compute cluster (HPC) [238]. Because the pruned VGG model remained frozen and the dataset is small, training was also conducted on a desktop machine using an NVIDIA RTX 2080 ti, with
some memory limitations. The need for the HPC was due to memory requirements and less so computational power. Multiple folds could not be stored in RAM. Further, for batch size, GPU memory was a factor.

At 870 TFLOPS, ManeFrame II has 354 nodes, 11,276 AVX2 Intel CPU cores, 275,968 accelerator cores, 120 TB in total memory, 100 Gb/s node interconnect bandwidth, and 2.8 PB of scratch space. The accelerator nodes include 36 NVIDIA P100 GPUs with 16 GB CoWoS HBM2 memory and 24 NVIDIA V100 GPUs with 32 GB of CoWoS HBM2 memory [238].

The desktop machine included a 12 core AMD Ryzen 9 3900X, 64 GB of DDR4 RAM, SSDs totaling 6 TB, and an NVIDIA RTX 2080 ti with 4352 cores and 11 GB GDDR6 memory.

9.9. Discussion

Overall, the performance of the model was comparable to an instructor for verifying gaze quality. It should be possible to deploy this model for applications such as (1) augmenting instructor observations or (2) training pilots to better scan for different maneuvers automatically in a real-time environment. The processing time required for the model is primarily due to the time needed for collecting gaze points. If implemented as a pipeline and primed with the initial observation window that slides overtime, the model could support a frame
rate of greater than 30 FPS. The actual time required for just the classification portion took, on average, less than 400 ms running on a 2018 Mac Book Pro. Therefore, the creation of an interface which, in real-time, displays the predicted gaze pattern quality for the pilot, is possible. This can assist in adjusting their technique during practice sessions or in mission execution.

In the same manner, pattern quality could be used by an instructor in a dashboard for a class of student pilots. Such a dashboard might increase the number of simultaneous pilots an instructor could effectively observe in a training setting. One limitation of the current model is that it can display the overall quality of the scan pattern, but it cannot display what corrective actions a pilot might take to increase the scan pattern quality. In future work, it may be possible to use gradient class activation mapping (Grad-CAM) to trace back to what portions of the input heatmap cause the model to have degraded performance [230].
From this knowledge, it should be possible to interpret the Grad-CAM output to instructions such as “Check your airspeed more often to improve scan quality”, “Keep your airspeed and wingman in your scan, you are fixated on the altitude indicator”. With this kind of approach, a multi-modal variant could be created that takes into account aircraft state, gaze pattern quality, Grad-CAM, and a student’s site picture informing a student how to improve upon this site picture and scan with specific inputs. As an example, on final approach, the system could say something like, “You are not keeping your airspeed in your scan, pitch down 2° for airspeed”. Further, there are several ways to solve a glide slope problem on approach. This kind of model could inform an optimal solution based on the student perspective — with repetition, potentially improving perceptual awareness [164].

Given the performance of the multi-task model, it is preferable compared to the single-task model. However, this does introduce some hurdles for deployment. For instance, in a real-time pilot feedback interface, the inference system would need to understand or be informed which flight phase the pilot was undertaking. Some of these maneuvers are easily categorized, such as using the “weight-on-wheels” signal of the aircraft to know when the aircraft is on the ground. Others, however, would require additional classification of the phase or require the instructor/pilot to select what phase or maneuver is currently being undertaken. While potentially minor, it does add an extra layer to a system that we prefer to be completely automatic.

The results of our research may be applied outside of the virtual environment, in actual flight. For an apples to apples comparison, Figure 9.14 depicts the data from a complete final approach for both the mixed-reality simulator and a real-life flight conducted on a C-17 military cargo aircraft. This example is an instance of gaze pattern data having been collected on a real flight. What has not been shown is how the scan pattern quality can be assessed automatically in real flight; there are several hurdles that will need to be overcome. The noise sources of sunlight, head/body movement from G-forces, and overall head movement in a actual flight could potentially reduce the accuracy of the model. In future work, deploying this model to real flight would need research into the noise sources and sensitivity of the
model to this additional noise. Even so, we hypothesize that the model could work in real flights because the simulations are high quality, such that noise sources from focal length and maneuver specific noise are already captured well by the model. Future work will investigate this more systematically.

Figure 9.14: Aggregate example heatmaps for final approach: (a) zoomed in heatmap of HUD, (b) Aggregate real-life flight on final approach in a C-17 aircraft [154]

9.10. Conclusions

In this research, we used convolutional neural networks to classify gaze or scan pattern quality for aviators in a multi-device, mixed reality aviation environment. We designed a human subjects experiment to inform the design and evaluation of these models with 40 subjects performing common flight maneuvers. We recruited three subject matter experts to rate the gaze patterns and analyzed their agreement, showing they have strong inter-rater reliability. Our multi-task convolutional neural network matched subject matter experts with greater than 93% average accuracy and strong multi-rater agreement, a $\kappa$ of .77.

This result suggests that gaze patterns for various flight maneuvers can be automatically classified into three levels of quality with reliable accuracy and in near real-time. This automated gaze classification may be of use in establishing the context of an aviator while they are learning a particular flight maneuver. The scope of our conclusions is limited to
gaze patterns in the scenarios from our experiments, but gaze classification for additional flight maneuvers or for other activities in other domains may also be applicable. We leave the investigation of gaze quality classification in additional flight maneuvers and other disciplines to future work.
Chapter 10

CONCLUSIONS

The overall goal of this research was to assist in the creation of a situation-aware inference system which can elucidate the human-machine state by inferring the internal states of a human subject. This could be accomplished through an evaluation of cognitive load, gaze scan patterns, and potentially situational awareness. We used deep learning and context-aware computing guided by several theories within the domain of psychology and flight. While we used the aviation domain as a use case, we believe the applications of this research are even more far reaching. An inference system that can perform assessments, reliably measure gaze, situational awareness, cognitive load of a pilot, or potentially any of number of effective-states in near real-time using multiple physiological sensors can ultimately assist instructors, students, and trained aviators, if not the greater community at large.

The capability of measuring human performance via biometric sensors is a research goal of many organizations, including the Department of Defence, gaming, cyber security, commercial aviation industries, and academic communities and institutions. Some applications include: (1) acceleration and amplification of knowledge acquisition and understanding of an exercise or skill through insights into the moment of skill mastery; (2) the relation of operational or academic student successes to their collected human performance data, which can shed light on the understanding of a subject’s susceptibility to different types of error that might be inferred; (3) real-time, self-assessment and personal development through feedback systems for a subject outside of student environments; and (4) adaptive user interfaces adjusting themselves based upon the inferred internal state of the user — situational awareness, attention, cognitive load, emotion, stress, engagement, etc.

Research within context-aware computing requires domain specific knowledge, [204]. This a priori knowledge and awareness serves to narrow the scope of context in order to make
it useful. The concept of using ubiquitous computing for augmented learning environments has been brought to the foreground on more than one occasion, [191, 204, 269, 282], where a plethora of sensing modalities have been used. Yvonne Rogers has described the classroom as an under-explored area which has the necessary ability to narrow the scope of the context required [205]. We believe the aviation and learning environment narrows the scope of context sufficiently needed for operable and actionable decisions about a pilot. The focus of this research used the practice of aviation as an a priori domain to enable the evaluation of several key internal human states — latent human context. We therefore used the various advances in deep learning such as multi-modal and multi-task learning to elucidate this context.

10.1. Contributions

The activities of this research involved three human-subject experiments that collected physiological measures relevant to aviators, including trained pilots, operators, and novice flyers. The data were collected while they underwent several scenarios, during both simulated and actual flight. An accurate, subject agnostic, multi-task gaze pattern classifier was constructed and demonstrated, as evidenced in Chapter 9. Accurate multi-modal and multi-task binary cognitive load classifiers have been successfully built, and which work across subjects and flight maneuvers, Chapter 8. The results are conclusive that cross-subject classification can be accomplished with greater than 81.0% accuracy. Our approach was validated using multiple physiological measures captured in real-flight. The most significant contributions of this research being: (1) a generalizable cognitive load classifier that works inter-subject and inter-task, and (2) a generalizable gaze scan pattern classifier that works inter-subject without the need for a bounding box.

In this research several contributions have been made and we briefly review them here. We introduced the concept of latent context as a third operative category of context (Section 2.3.2). We designed and carried out a human subjects experiment for aviators in a variety of flight scenarios, as reviewed in Chapters 6 and 7. Biometric data, Chapter 5, were
collected for five modalities across six low and high workload maneuvers. We proposed a novel subjective label alignment scheme using subjective ratings for inter-task cognitive load classification, discussed in Section 8.2.1. We investigated three competing machine learning architectures for the classification of cognitive load: (1) a multi-modal feature engineered random forest classifier, (2) a fixed-length input deep learning classifier, and (3) a variable-length input deep learning classifier. The deep learning models took advantage of generative learning, multi-modal learning, multi-task learning, and X-Vector like architectures. Finding, that a multi-modal, binary model can automate the evaluation of cognitive load with an accuracy of over 81.0% agnostic to the subject, and to flight maneuver. We validated our cognitive load deep learning approach with real-world data from five test pilots collected over two military test and evaluation flights on a C-17 aircraft (CL > 90.0% accurate with a few caveats). We proposed methods for novel gaze data augmentation specific to pilot scan patterns that increase the robustness of trained machine learning models (Section 9.5). We investigated two competing, convolutional neural network architectures: a task agnostic model and a multi-task model (Section 9.4) for the classification of pilot gaze patterns. We found we can evaluate the architectures with K-fold cross-validation, achieving greater than 93.0% average test accuracy compared to instructor observation (Section 9.6). We further analyzed gaze scan pattern inter-rater reliability amongst raters and classifiers finding strong agreement; and in some cases, machine-to-human IRR is higher than human-to-human IRR, Section 9.3.

10.2. A Way Forward

Going forward two important research questions that have yet to be answered include:

*Can we accurately classify situational awareness using multiple modalities including physiological measures, models of psychology, aircraft state, among others?*

*Can the use of physiological measures for personalized learning be demonstrated, and to what degree of acceleration can be provided?*
In order to build a situation-aware system that can reliably evaluate situational awareness and cognitive load of a pilot, we first needed to collect physiological data where human-subjects complete various tasks of varying difficulty. With this data, multiple classifiers were built including cognitive load and gaze scan patterns. Using multiple psychology models — CLT 4.2, SAGAT 4.1, Yerkes-Dobson bell curve [67] and Hancock and Warm’s dynamic model of stress and sustained attention [108] — we believe an assessment about situational awareness may be made. There is great potential to maximally increase performance and quality of training through the use of the Yerkes-Dobson bell curve [280], Hancock and Warm’s dynamic model of stress and attention [108], and the Cognitive Load Theory, Section 4.2.

Work in the future should involve the broadening and/or completion of the critical components for the situation-aware system, which include increasing the sensitivity of the cognitive load classifier — much more data is necessary. The addition of flushed out stress, frustration classifiers — NASA-TLX sub-scales may be useful as proxies for stress and frustration. Applying these measures of cognition to both the Yerkes-Dobson and Hancock and Warm bell curves. A key issue with the classification of cognitive load is labeling. And development of a situation awareness performance index given Endsley’s work, [78–81].

Interestingly, through this research Scielzo et al. has already established a relationship between situation awareness and analogous machine learning metrics — pilot gaze scan pattern and mental workload classification. Specifically, the three level of situation awareness (perception, comprehension, and projection) are compared to the analogous ML metrics, performance, and experience through statistical hypothesis testing. This important work should be repeated with our trained classifiers [229].

An important research area is cognitive load subjective rating alignment. This research showed that using the assumed difficulty of a task is limited, Section 4.3. Given the influence of the acceleration modality on model accuracy, Section 8.4, modeling more levels of load by examining activity, such as in Gray’s work on inceptor workload [99], may help. The trouble caused by attempting to cut through the bias in subjective ratings highlights the
critical need for a highly sensitive objective workload measure. Cognitive load subjective rating alignment is an open research problem that must be investigated further.

In seeking to measure human performance objectively, we ultimately strive to understand the internal context of a person during any task, operation, or safety-critical activity. If we can understand the important characteristics of that person’s experience—this latent human context—we increase the informational bandwidth and enrich the interaction between the human and the machine.
Appendix A

BEDFORD WORKLOAD SCALE

Figure A.1: Bedford Workload Scale
Appendix B

NASA TASK LOAD INDEX RATING

Subject ID: ___________________ Task ID: ___________________

RATING SHEET

MENTAL DEMAND
Low | High

PHYSICAL DEMAND
Low | High

TEMPORAL DEMAND
Low | High

PERFORMANCE
Good | Poor

EFFORT
Low | High

FRUSTRATION
Low | High

Figure B.1: NASA Task Load Index Rating Sheet Unweighted (RTLX)
<table>
<thead>
<tr>
<th>Title</th>
<th>Endpoints</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MENTAL DEMAND</td>
<td>Low/High</td>
<td>How much mental and perceptual activity was required (e.g., thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex, exacting or forgiving?</td>
</tr>
<tr>
<td>PHYSICAL DEMAND</td>
<td>Low/High</td>
<td>How much physical activity was required (e.g., pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slow or brisk, slack or strenuous, restful or laborious?</td>
</tr>
<tr>
<td>TEMPORAL DEMAND</td>
<td>Low/High</td>
<td>How much time pressure did you feel due to the rate or pace at which the tasks or task elements occurred? Was the pace slow and leisurely or rapid and frantic?</td>
</tr>
<tr>
<td>PERFORMANCE</td>
<td>good/poor</td>
<td>How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals?</td>
</tr>
<tr>
<td>EFFORT</td>
<td>Low/High</td>
<td>How hard did you have to work (mentally and physically) to accomplish your level of performance?</td>
</tr>
<tr>
<td>FRUSTRATION LEVEL</td>
<td>Low/High</td>
<td>How insecure, discouraged, irritated, stressed and annoyed versus secure, gratified, content, relaxed and complacent did you feel during the task?</td>
</tr>
</tbody>
</table>

Figure B.2: NASA Task Load Index Rating Definitions
### Appendix C

**SAGAT QUESTIONNAIRE**

#### SAGAT Stop 1-A: Freeze sim after descending pull for offset
- What is your throttle setting? (full, 3/4, 1/2, 1/4, idle)
- What is your target aspect? (hot 0-20, flank 30-60, beam 70-110, cold 120)
- What is your closure rate?
- What is the impact of the closure rate: are you slower, are you co-airspeed, or are you faster?
- What is your airspeed?
- What is your altitude?
- What is your general direction? (Cardinal)
- What is your distance to the bandit?
- What is the bandit's altitude?
- How many degrees is your offset (antenna train angle ATA)?
- How far off are you from desired airspeed?
- How off are you from desired pitch attitude?
- When will you reach desired altitude?
- When will you reach desired offset heading?

#### SAGAT Stop 2-A: Stop 5 nm from bandit
- What will be your rollout intercept positioning? (sucked, acute, or in position)
- What is the impact of the closure rate: are you slower, are you co-airspeed, or are you faster?
- What is your target aspect? (hot 0-20, flank 30-60, beam 70-110, cold 120)
- What is your airspeed?
- What is your altitude?
- What is your heading?
- What is your separation distance?
- How fast is the bandit going?
- How fast are you closing?
- What is your vertical separation?
- Given your current bank angle, what is the impact on the intercept? Are you early, on, or late?
- When will you be within 1 mile?
- When will you be within 0.1 mile?

---

Figure C.1: Alpha Questionnaire
**SAGAT Stop 1-A: Freeze sim after descending pull for offset**

- When will you reach desired offset heading?
- How many degrees is your offset (antenna train angle ATA)?
- When will you reach desired altitude?
- How off are you from desired pitch attitude?
- What is the bandit’s altitude?
- What is your distance to the bandit?
- What is your general direction? (Cardinal)
- What is your altitude?
- How far off are you from desired airspeed?
- What is the impact of the closure rate: are you slower, are you co-airspeed, or are you faster?
- What is your throttle setting? (full, 3/4, 1/2, 1/4, idle)
- What is your target aspect? (hot 0-20, flank 30-60, beam 70-110, cold 120)
- What is your closure rate?
- What is your airspeed?

**SAGAT Stop 2-A: Stop 5 nm from bandit**

- When will you be within 1 mile?
- How fast is the bandit going?
- What is your heading?
- What is your separation distance?
- What is the impact of the closure rate: are you slower, are you co-airspeed, or are you faster?
- When will you be within 0.1 mile?
- What is your vertical separation?
- What is your attitude?
- How fast are you closing?
- What is your airspeed?
- What is your target aspect? (hot 0-20, flank 30-60, beam 70-110, cold 120)
- What will be your rollout intercept positioning? (sucked, acute, or in position)
- Given your current bank angle, what is the impact on the intercept? Are you early, on, or late?

Figure C.2: Bravo Questionnaire
Appendix D

BIOMETRIC CAPTURE SOFTWARE

A custom application was created to store and view biometric data from the E4 wrist-band in real time. The biometric capture software developed has the following capabilities: E4 sensor management, real-time data stream capture and formatting, data stream visualizations (pinch and zoom), bookmarking, secondary task data capture (Oddball, further discussed in the L3Harris Feasibility Study Methodology, Section 6.2). The software was written in swift and objective-c for use on the Apple iPad or iPad mini, iOS 12.0 or later.

D.1. JSON Format

Streamed data to the iPad was stored via the following JSON format:

Onboard storage:

{

“WristBiometric” : {

“ExperimentID” : “user defined string”,

“SubjectID” : “user defined string”,

Automatic annotation of E4 placement, accuracy is limited,
useful only as a data analysis guide

“OnWrist” : {

“timestamps” : [string representation of floats, UTC],

“values” : [“string representation of boolean values”]

},

}
32 Hz

“Acceleration” : {
    “timestamps” : [string representation of floats, UTC],
    “x” : [string representation of floats, g],
    “y” : [string representation of floats, g],
    “z” : [string representation of floats, g]
},

4 Hz

“Temperature” : {
    “timestamps” : [string representation of floats, UTC],
    “values” : [string representation of floats, °C]
},

The measurement unit for BVP (PPG) is fraction of nano Watt. This is basically a difference of light absorption observed by pulse oximeter. 64 Hz

“BVP” : {
    “timestamps” : [string representation of floats, UTC],
    “values” : [string representation of floats, BVP (nW)]
},

Electro-dermal activity (EDA), 4Hz, micro Siemens (µS)

“EDA” : {
    “timestamps” : [string representation of floats, UTC],
    “values” : [string representation of floats, µSiemens]
},
Heart rate = 60 / inter beat interval

"IBI" : {
  "timestamps" : [string representation of floats, UTC],
  "values" : [string representation of floats, seconds]
},

Commonly known as mark or event marking; allows for real-time annotation by marking the time when the button was pressed

"Mark" : {
  "timestamps" : [string representation of floats, UTC],
  "values" : [string representation of integers, mark count]
}
}
D.2. Screen Shots

(a) Main view: before connection

(b) Main view: connected with data

(c) Photoplethysmography Chart View

(d) Acceleration Chart View

(e) Electrodermal Activity Chart View

(f) Skin Surface Temperature Chart View

Figure D.1: Capture Software Screen Shots
BIBLIOGRAPHY


[56] DAWSON, M., SCHELL, A., AND FILON, D. The electrodermal system.


[77] Empatica Support. What should I know to use the PPG/IBI data in my experiment?, 2018.


[110] Hanson, T. L3 Introduces First-Ever High-Fidelity, Mixed Reality Deployable Training Simulator, Nov. 2018.


175


183


[263] Valerie, A., Huemer, Hayashi, M., Renema, F., Elkins, S., McCandless, J. W., and McCann, R. S. CHARACTERIZING SCAN PATTERNS IN A SPACECRAFT COCKPIT SIMULATOR: EXPERT VS. NOVICE PERFORMANCE.


[270] WANGWIIWATTANA, C. RGB Image-Based Pupillary Diameter Tracking with Deep Convolutional Neural Networks.


