



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**MITIGATING INSIDER THREATS IN VIRTUAL
ENVIRONMENTS WITH DECEPTION DETECTION**

by

Jamie Francona

June 2021

Thesis Advisor:

Co-Advisor:

Vinnie Monaco

Cynthia E. Irvine

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 2021	3. REPORT TYPE AND DATES COVERED Master's thesis	
4. TITLE AND SUBTITLE MITIGATING INSIDER THREATS IN VIRTUAL ENVIRONMENTS WITH DECEPTION DETECTION			5. FUNDING NUMBERS	
6. AUTHOR(S) Jamie Francona				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING / MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.			12b. DISTRIBUTION CODE A	
13. ABSTRACT (maximum 200 words) Insider threats pose a severe risk to DOD networks and the missions they support. With trusted access, insiders can manipulate critical and sensitive cyber systems. These threats can be difficult to detect, given their trusted access. There is a need for early indications and warnings of deceptive activity so that the damage from these malicious actors can be stopped or limited as quickly as possible. Current deception detection capabilities include behavioral and physical biometrics, but these techniques do not address unencountered users. This study researches the merit of using human-computer interaction (HCI) features for a deception detection capability. With data collected in an online survey, machine learning is used to classify deceptive or potentially deceptive online behavior using keyboard and mouse movement. This study demonstrates the potential for utilizing (HCI) as an indicator for deception and offers the possibility of detecting deception in unencountered users. It expands the effectiveness of early insider threat detection by demonstrating the ability to classify concealed or deceptive user activity without the need for a user-specific model created from per-user historical data.				
14. SUBJECT TERMS Department of Defense, DOD, Naval Postgraduate School, NPS, Concealed Information Test, CIT, human-computer interaction, HCI, Myers-Briggs Type Indicator, MBTI, false positive, FP, false negative, FN, comma-separated values, CSV			15. NUMBER OF PAGES 125	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

**MITIGATING INSIDER THREATS IN VIRTUAL ENVIRONMENTS
WITH DECEPTION DETECTION**

Jamie Francona
Lieutenant, United States Navy
BS, U.S. Naval Academy, 2016

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

**NAVAL POSTGRADUATE SCHOOL
June 2021**

Approved by: Vinnie Monaco
Advisor

Cynthia E. Irvine
Co-Advisor

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Insider threats pose a severe risk to DOD networks and the missions they support. With trusted access, insiders can manipulate critical and sensitive cyber systems. These threats can be difficult to detect, given their trusted access. There is a need for early indications and warnings of deceptive activity so that the damage from these malicious actors can be stopped or limited as quickly as possible. Current deception detection capabilities include behavioral and physical biometrics, but these techniques do not address unencountered users. This study researches the merit of using human-computer interaction (HCI) features for a deception detection capability. With data collected in an online survey, machine learning is used to classify deceptive or potentially deceptive online behavior using keyboard and mouse movement. This study demonstrates the potential for utilizing (HCI) as an indicator for deception and offers the possibility of detecting deception in unencountered users. It expands the effectiveness of early insider threat detection by demonstrating the ability to classify concealed or deceptive user activity without the need for a user-specific model created from per-user historical data.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1	Introduction	1
1.1	Motivation and Problem Statement	1
1.2	Contributions	3
1.3	Thesis Organization	3
2	Background and Related Work	5
2.1	Deception Detection	5
2.2	Behavioral Biometrics	6
2.3	Related Work	11
3	Survey Methodology	19
3.1	Experiment Overview	19
3.2	Questionnaire Format	20
3.3	Survey Launch	25
4	Data Collection	31
4.1	Web Logging	31
4.2	Data Pre-processing	31
5	Deception Detection Methodology	39
5.1	Deception Cues in HCI	39
5.2	Feature Extraction	44
5.3	Models	45
5.4	Splitting Data into Subsets	47
6	Results and Discussion	49
6.1	Model Accuracy and Results Using a Random Forest Classifier	49
6.2	Model Accuracy and Results Using the Extra Trees Classifier	58

6.3	Prediction Examples	59
6.4	Discussion	88
7	Conclusions and Future Work	91
7.1	Future Work	91
7.2	Contributions and Conclusion	92
	Appendix A Dictionary Mapping of Web-Logged Events	95
	Appendix B Complete List of Survey Questions	99
	List of References	101
	Initial Distribution List	105

List of Figures

Figure 2.1	Changes in velocity for questions with varying truth values. . . .	14
Figure 2.2	Mouse movement and trajectory changes of liars.	16
Figure 3.1	How deceit alters mouse trajectory	21
Figure 3.2	Mouse movement trajectory on expected and unexpected questions	22
Figure 3.3	Survey e-mail invitation	27
Figure 3.4	Survey landing page	28
Figure 3.5	Survey Honor Code	29
Figure 4.1	Raw data header	32
Figure 5.1	Velocity histogram with 100 Bins	40
Figure 5.2	Change in Time Histogram with 40 Bins	42
Figure 5.3	Angle Histogram with 50 Bins	43
Figure 5.4	Extract features function	44
Figure 5.5	Code to run extract_features() on the dataframe	45
Figure 6.1	Individual user accuracy of binary classification on entire dataframe of red versus green questions	61
Figure 6.2	Individual user accuracy of binary classification on entire dataframe of red versus green questions	62
Figure 6.3	Individual per-user accuracy results on test set of red versus green radio button questions	63
Figure 6.4	Mouse trajectory effect on <i>angle</i> measurement	67

Figure 6.5	Velocity histograms for a green and red radio button response from survey 10	69
Figure 6.6	Velocity profiles for a green and red radio button response for survey 10	70
Figure 6.7	Angle histograms for a green and red radio button response for survey 10	71
Figure 6.8	Angle profiles for a green and red radio button response for survey 10	72
Figure 6.9	Velocity histograms for a green and red radio button response for Survey 13	73
Figure 6.10	Velocity profiles for a green and red radio button response for survey 13	74
Figure 6.11	Angle histograms for a green and red radio button response for survey 13	75
Figure 6.12	Angle profiles for a green and red radio button response for survey 13	76
Figure 6.13	Change in time histograms for a green and red radio button question for survey 13	77
Figure 6.14	Change in time profiles for a green and red radio button response for survey 13	78
Figure 6.15	Velocity histograms for a green and red radio button question for Survey 28	79
Figure 6.16	Velocity profiles for a green and red radio button response for Survey 28	80
Figure 6.17	Angle histograms for a green and red radio button response for survey 28	81
Figure 6.18	Angle profiles for a green and red radio button response for survey 28	82
Figure 6.19	Change in time histograms for a green and red radio button response for survey 28	83

Figure 6.20	Change in time profiles for a green and red radio button response for survey 28	84
Figure 6.21	Velocity histograms for a green and red radio button response for survey 30	85
Figure 6.22	Velocity profiles for a green and red radio button response for survey 30	86
Figure 6.23	Angle histograms for a green and red radio button response for survey 30	87
Figure 6.24	Angle profiles for a green and red radio button response for survey 30	88

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Table 2.1	Related works	12
Table 3.1	Color coded questions	23
Table 5.1	Data subsets for modeling	48
Table 6.1	Accuracy results of binary classifications of entire dataframe using only velocity with Random Forest Classifier	50
Table 6.2	Accuracy results of binary classifications of radio button questions only using velocity with Random Forest Classifier	51
Table 6.3	Accuracy results for binary classifications of text-based questions using velocity only with Random Forest Classifier	52
Table 6.4	Accuracy results of binary classifications of entire dataframe using Angle with Random Forest Classifier	53
Table 6.5	Accuracy results of binary classifications of radio button questions using angle with Random Forest Classifier	54
Table 6.6	Accuracy results of binary classifications using change in time with Random Forest Classifier	55
Table 6.7	Accuracy results for binary classifications of radio button questions using change in time with Random Forest Classifier	56
Table 6.8	Accuracy results for binary classifications of radio button questions using velocity and angle with Random Forest Classifier	56
Table 6.9	Accuracy results for binary classifications of radio button questions using velocity and delta time with Random Forest Classifier	57
Table 6.10	Accuracy results of binary classifications of radio button questions using velocity, angle, and change in time with Random Forest Classifier	58
Table 6.11	Accuracy results of binary classifications using velocity, angle, and change in time using Extra Trees Classifier	58

Table 6.12	Accuracy results of binary classifications using velocity, angle, and change in time using Extra Trees Classifier	59
Table 6.13	Accuracy results of binary classifications of radio button questions using velocity and change in time with Extra Trees Classifier . . .	59
Table 6.14	Summary statistics of correctly classified radio button surveys . .	65
Table 6.15	Standard deviations on features of correctly classified radio button surveys	68
Table A.1	Dictionary mapping of computer keyboard inputs to Numeric Codes.	95

List of Acronyms and Abbreviations

DOD	Department of Defense
NPS	Naval Postgraduate School
CIT	Concealed Information Test
HCI	human-computer interaction
MBTI	Myers-Briggs Type Indicator
FP	false positive
FN	false negative
CSV	comma-separated values

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I would like to thank all of my teachers and professors, past and present, who have played a piece in helping me to this point, and who undoubtedly will continue helping me learn and grow throughout life. A special thank you to my thesis advisers, Vinnie and Cynthia, who endured (and probably suffered) through many virtual meetings throughout this process. Thank you for encouraging me, believing in me, and pushing me to do something well outside of my comfort zone. Thank you for the constant support, for keeping me on track, and for being a delight to work with from start to finish.

Thank you especially to my family, friends, and cohort for not only keeping me sane, but making this a truly wonderful experience and time in my life. Special thank you to my mom, Jacque, and my husband, Landon, who were and always will be my biggest cheerleaders. Endless thank yous to Landon, who never acted bored when reading chapter after chapter, while trying to relax on the couch. Thank you for encouraging me and for picking up the slack while I worked towards the finish line.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:

Introduction

1.1 Motivation and Problem Statement

Department of Defense (DOD) networks maintain a myriad of sensitive information systems that must be secured and protected. Insider threats, along with malicious activity aimed at gaining entry into a sensitive network, pose a severe risk to DOD networks and their associated missions based on the trusted access that personnel are granted for use of these critical and sensitive systems and networks. There are few mitigation or detection measures currently in place to monitor insider threat or deceptive activity on a network beyond basic routine password or credential checks. Although insider threat has been active area of research on hosts, network insider threat detection has received little attention. This type of work primarily addresses two categories of insider threats: naive users and skilled impersonators at the onset. A naive user is someone who believes they can simply falsify answers to sensitive security questions without ramification. On the other hand, a skilled impersonator may be someone who provides information at the time of account creation or before trust is established, attempting to gain unauthorized access to one or more systems. Other methods of detection in place are based on detecting deviations from known behavior and therefore rely on a user's historical data as a necessary comparison. For this reason, it becomes even harder to detect a threat from a user who is new to a network, and therefore does not have historical information to be compared to. Once an insider threat has gained access to a network, it is exponentially more difficult to manage or track them. If unmanaged, these threats could permeate all levels of DOD networks, affecting everything from administrative files to combat readiness and national security.

This research aims to better understand the connection between deceptive insider threat activity and human-computer interaction (HCI) in order to determine if elements of a user's HCI can be utilized as a means of virtual threat detection. Specifically, this study aims to eliminate the need for a user-specific model, thus attempting to prevent profile-less users from gaining access and remaining undetected on critical networks. This research examines how active deception manifests in HCI when users participate in an online survey. Using

the survey, we specifically explore keyboard interaction and mouse movement, with the aim of discovering deception through machine learning algorithms.

More specifically, this research focuses on three main HCI features to view deception detection: *velocity*, *angle*, and the change in time (*delta_time*). These features are used individually and in combination to determine more reliable and consistent indicators of deception. The two machine learning algorithms used for modeling in this study are the Random Forest Classifier and the Extra Trees Classifier.

1.1.1 Operational Relevance

If cyber insider threat detection can be automated to any degree, the DOD's overall awareness of potential threats on sensitive networks would increase. Determining what threat or malicious behavior looks like in terms of HCI, and how it varies compared to typical computer use, will enable a speed up of indications and warnings of network threat activity, followed by the potential to limit threat access in order to mitigate malicious exposure. The ability to determine deceptive behavior in first-time users not only benefits insider threat detection within the DOD, but can be utilized in many other workplaces to protect personnel and intellectual property. The ability to detect deception through a user's keyboard and mouse interactions offers the potential to limit dishonest behavior within the work force, thereby creating additional barriers to entry for legitimate personnel who may attempt to gain access to information outside of their realm, despite having proper access and clearance to a network. Protecting the force against malicious threat actors and deception requires a whole-of-force and defense-in-depth approach. The DOD must address a high level of threat activity aimed at harming and degrading its effectiveness everyday. This threatening behavior can originate from adversary countries, but can also originate from within the country, or even within the force. Utilizing HCI to help narrow and defend along these threat axes would create a more robust and secure DOD. The more secure a force is from insider threat or malicious activity, the more it can focus on harnessing warfighting capabilities to defend the country and her interests.

1.2 Contributions

The following list highlights the largest contributions of this study to the associated field of research.

- Validate previous work discussed in Chapter 2 on how “unexpected” questions and orienting responses change when a user is honest versus deceptive.
- Determine the effectiveness of *velocity*, *angle*, and *delta_time* on deception detection and on a non-user specific model accuracy.
- Compare case studies of certain individuals in truth-telling versus deceptive scenarios in order to view *velocity*, *angle*, and *delta_time* histograms and profiles.
- Gain insights into how deception may manifest through specific features such as *velocity*, *angle*, and *delta_time*.
- Provide insight into the use of HCI features for deception detection in a non-user specific model.
- Analyze per-user accuracy to determine how non-user specific models vary from user to user
- Model and analyze classifiers that detect deception based on web-logged survey information from research participants

1.3 Thesis Organization

Chapter 2 discusses background information related to this study, including deception detection at large and behavioral biometrics. The chapter concludes with a summary of related work and what this study aims to contribute beyond this previous work. Chapter 3 discusses our experiment, the creation and delivery of the survey, the format of the questionnaire, and the population surveyed. Chapter 4 follows with a discussion regarding the web logging code that was created in order to capture the specified HCI features. The chapter concludes with details on how the data was cleaned and pre-processed prior to modeling. Chapter 5 discusses model creation, including feature selection and extraction, along with the models that were used and how the data was split into subsets. Chapter 6

discusses the results of various models in depth. Additionally, case studies are examined to view the *velocity*, *angle*, and *delta_time* histograms and profiles of individual users. The chapter concludes with a discussion of how certain factors may have influenced the study and its associated outcomes. The final chapter, Chapter 7, discusses the overall contributions of this study and the future work that our study can inform.

CHAPTER 2:

Background and Related Work

This chapter begins by familiarizing the reader with key concepts related to our research, beginning with deception detection. It then explores topics such as behavioral biometrics, and addresses some key shortcomings of using the aforementioned to support security endeavors within and throughout systems. After a brief introduction to biometrics, human-computer interaction (HCI) is explained and introduced as a possible means of detecting deception. Previous and related work is detailed as the starting point for our study. The chapter concludes with a brief discussion of what our research aims to add to previous studies.

2.1 Deception Detection

Deception, at its root, occurs when a person uses “messages or signals knowingly and intentionally” in order to “foster a false belief or conclusion by another” [1]. Deception can manifest in many ways, including, but not limited to, counter-factual responses, information concealment, or even by deceiving while still telling the truth [2]. Deception is not simply lying or withholding information, but rather encompasses a broad spectrum of actions with fluctuating motivations. On one side of the continuum, deception can be used in the form of little white lies, motivated by the need for surprise, in interpersonal relationships. An example on this end of the spectrum would be a small white lie surrounding a holiday surprise, such as a surprise birthday party or a surprise Christmas gift. While the intent is benign and the consequences minimal, this represents just one end of the spectrum. In other cases, the motivation to deceive is ill-intended, as occurs when information is falsified, intentionally misleading, or simply withheld [2]. The intent behind the deceptive actions often determines the severity of its consequences. For example, if the intent to deceive is motivated by evil, such as theft or fraud, the consequences of these actions, successful or not, are generally much more severe than those for a birthday surprise gone wrong. Motivation for deception detection exists in nearly every sector of life. Unfortunately, detecting deception can be time-consuming, can require expensive and unusual tools and equipment, can be obtrusive, and can often result in false positives (FPs) and false negatives (FNs).

Deception, on some scale, occurs in every facet of life, from interpersonal relationships to business ventures, with varying degrees of consequence and risk. Exorbitant amounts of money are lost annually by organizations that fail to appropriately monitor and respond to information concealment, which can potentially lead to security incidents [3]. Despite the financial burden that these deceptive acts can place on organizations, many corporations still report insufficient control methods in place to detect and respond appropriately and efficiently to concealed or misleading information attacks [3]. Within some organizations, deceptive actions can pose even graver risk than monetary loss, such as in the DOD or the Department of Homeland Security. In these instances, deception threats have the potential to permeate networks at all classification levels. This can increase the risk that sensitive or even highly classified information may be altered or exfiltrated to unauthorized personnel who lack the proper clearance and need-to-know. Deceptive threats to these entities, while simultaneously incurring financial burdens, could put the nation's combat readiness and security at risk [4].

2.2 Behavioral Biometrics

A biometric, in general, refers to a measurable characteristic that is unique for each individual and may be used to verify or authenticate a person's identity [5].

Detecting deceptive behavior by individuals largely starts with the ability to detect change, either in the manner in which something is done, or their body's natural response to an event. These personal changes can be detected through numerous avenues, most commonly physiologically or behaviorally, and can be utilized as biometrics. Physiological changes can be revealed through measurements such as blood pressure, heart rate, sweat, and muscle action potentials [6]. The polygraph-based Concealed Information Test (CIT) that is used by the DOD relies on detecting physiological transformations, and more specifically, electro-dermal activity and responses. Electro-dermal activity is the measurement of change in a skin's response to small electrical currents. While these involuntary changes result in noticeable and measurable bodily fluctuations, they are not the only changes that take place in the deceiver. In addition to changes at the physiological level, there are also noticeable behavioral changes that take place, such as variation in the pitch of a person's voice, cognitive performance changes, as well as facial expressions and changing body language [6]. These observable changes can be called behavioral biometrics, so long as they can uniquely

characterize an individual. Such biometrics often have advantages over physiological metrics. Utilizing a passive and transparent behavioral biometric does not require probing or extra interaction from the user, thereby limiting the footprint and obstruction imposed on them, and thus providing a significant advantage over physiological biometrics [6]. In fact, subjects may be unaware that these biometrics are even being collected at times.

While physiological and behavioral biometrics are valid approaches to identifying changes in a user's state, either emotional or physical, some approaches are not passive and can be quite cumbersome to implement [7]. In order to successfully detect biometric changes, specialized tools, which are not commonly available, can be necessary. Based on these key shortcomings, research in behavioral biometrics has largely shifted focus to the use of HCI as a means of collecting personal biometrics for a multitude of purposes, such as authentication, authorization, and deception detection.

Biometrics are often used for security purposes. For example, more recent versions of Apple's iPhone harnesses facial recognition technology for password authentication in lieu of typical numerical password entry, while older versions utilize fingerprint technology to perform similar functions. Depending on the end goal, the quality of uniqueness and accuracy of a biometric characteristic may vary. For example, when utilizing a biometric, such as fingerprint or facial recognition technology, for authentication to gain access to a personal device, the biometric is used in a simple one-on-one comparison. The fingerprint or face in question is only compared to the owner's stored hash of the fingerprint or face in order to authenticate the user. However, the store of biometrics used for identification may be many orders of magnitude larger. For example, databases could house millions of faces or fingerprints from which to compare, making the problem set more challenging. Expanding databases inevitably increase the risk for an incorrect match, or FPs, due to what is called the "doppelganger effect" [8]. Along with the potentially increasing FP rate comes an increasing moral dilemma. For example, a small technology start-up, Clearwater AI, utilizes facial recognition technology that scrapes all available social media outlets for photos, garnering people's personal information for exploitable purposes [8]. Depending on who obtains this type of technology, a wide array of privacy issues are raised. As with any form of authentication, the use of physical biometrics is not immune to adverse side effects and improper usage.

While both physical and behavioral biometrics can be used for similar purposes, it is important to differentiate between the two. Both iris and fingerprint scanning are considered physical attributes. On the other hand, behavioral biometrics are based on actions performed by a person, and are generally considered much more difficult to impersonate [5], thereby making them marginally less vulnerable to attack.

Behavioral biometrics refer to anything that a person does, such as the way they walk or talk, that can uniquely identify them. The primary reason that behavioral characteristics and responses may be difficult to mimic stems from the fact that they can be involuntary. Contrary to scientists' original understanding surrounding perception and action, recent studies deduce that decision processes are not entirely discrete, and therefore decision space can be shared between subsystems prior to decision completion [9]. This complex and continual relationship between the brain and the body's movement delivers deeper understanding of the inner workings of cognitive processes [10]. When a person attempts deceit, their cognitive load, when compared to telling the truth, increases, as the brain is required to process several sub-systems simultaneously to deal with items such as the morality of deception, the justification for said deception, and how to avoid being exposed [11]. This increased cognitive load is illustrated in subconscious physiological deviations, which manifest in certain action responses [12]. Therefore, when a person is asked a question and intends to lie, their brain concurrently processes both the lie that they desire to tell and what they understand as the ground truth. In other words, there tends to be a "significant *a priori* bias toward "truth," and this bias must be overcome before a "false" or deceptive response can come to fruition" [9]. The cognitive load that is placed on the intended liar alters their performance in a way that may be noticeable to observers. These physiological deviations can produce certain side effects, such as hesitation or reconsideration, that alter a person's movement away from a person's originally intended trajectory [13]. The trajectory deviation of a computer mouse opens up the possibility for alternative ways to detect deception.

2.2.1 Human-Computer Interaction

HCI can describe any feature that results from interaction between a user and an input or output to a computer, most commonly the mouse, keyboard, or more recently, touch screens. In addition to the human's interaction with the computer, the computer's interaction with the human and the resulting influence on their state is also considered HCI [14].

Interaction is much more involved than simple dialogue between a computer and a user, and includes transmission considerations, experience, control, and others, that dictate items such as channel performance and feedback, all of which can drastically affect a user's experience and mood, thereby affecting the user's reactions [15]. These manifested reactions can be collected through certain HCI functions and analyzed in an attempt to determine something about the user.

The way in which a human interacts with a computer can provide ample information about a person. In many ways, how humans interact with machines mirror their interactions with each other, and, therefore, can provide insight into interesting social questions. [6]. Because of the vast potential for data that HCI provides, many researchers have turned their attention to analyzing aspects of HCI and how computer interactions change from user to user depending on a person's personality traits, moods, etc. The differences in how users interact with computers can be used to profile and identify individuals, as well as determine factors such as emotional state and deceptive behavior [16].

Keyboards, mice, and touch-pads are common peripheral devices that can be used for analysis. Different measurements can be extracted from each of these devices in order to undergo further analysis. For example, mouse movement can be segmented into features such as velocity, acceleration, or trajectory. Likewise, inputs to the keyboard can be measured in a number of ways. Common keyboard features include dwell time, which is the duration of time that a key remains depressed, and key latency or flight, which is the time between one key being released and the next being depressed [7]. Analysis of the relationships between these HCI measurements can be used to gain information that not only helps explain who a user is, but may also explain why a user acts in a particular way [17].

Every time a person operates a computing device such as a personal computer, tablet, or cellular phone, HCI occurs, and chances are, users have consented to allowing researchers to use those analytics to better understand the human mind. Motivation for studying HCI extends well beyond computer scientists. Many sectors benefit from the information gleaned from HCI, particularly advertisers, businesses, and security developers.

2.2.2 User Profiling

As discussed in Section 2.2, user identification or authentication requires enrollment in a database. A key difference in user profiling is that it can be utilized on novel individuals based on observations from a general population, thereby making historical data on a specific user irrelevant. Although user profiling provides this advantage, it also introduces a variety of problems. User profiling is done in many ways and for many reasons. A common example of profiling that has negative connotations is the notion of the Transportation Security Administration profiling personnel going through airport security in order to identify people that pose a threat to security, such as national and international terrorists. This sort of user profiling can be based on body language or physical characteristics and attire, and it is largely subjective. This type of subjective user profiling can be prone to both false positives, which impose inconvenience and embarrassment on travelers, or, in the case of false negatives, can put airports and people in grave danger. In this scenario, a FP represents a mundane passenger who is flagged as a threat, despite no actual threat being present. On the other hand, a FN in this scenario represents a legitimate threat slipping through security without being flagged as a threat at all.

Another common type of profiling is based on personality traits. Personality assessments that categorize people based on predispositions are common. These types of assessments are commonly used by colleges or employers to better place students and employees in roles that they might find a propensity for success. For example, the MBTI defines personality traits based on four criteria that constitute opposing preferences, such as extroversion or introversion, or analytic versus emotional intelligence [18]. At the culmination of the personality test, a person can be described by a four-letter acronym, which is used to explain how their temperament impacts their behavior and actions [18]. For example, the most common MBTI personality type is the ISFJ, or the “Defender,” which accounts for approximately 13.8% of the population and defines someone who has the following personality traits: introverted (I), observant (O), feeling (F), and judging (J) [19] [20].

Since the examples above describe profiling that is prone to both subjectivity and the self-fulfilling prophecy effect, researchers seek objective measures. They are eager to use HCI features to profile users as discussed in Section 2.2.1, thus removing some human bias by trading subjectivity for computer analysis and the application of artificial intelligence techniques. Unfortunately, machine learning systems and algorithms can still be prone to

bias, creating models that “unintentionally produce data that encode gender, ethnic and cultural biases” [21]. Aspects of HCI can be used not only to identify and/or verify persons, but can also be used to classify people based on age, emotions, gender, etc. For example, the way in which a person types, to include all features that describe that person’s interaction with a keyboard, can be unique enough to identify not only the aforementioned characteristics, but the users themselves [7].

2.3 Related Work

Our research draws on related works that discuss user-specific models for profiling and change detection, specifically deception detection utilizing HCI.

Table 2.1 summarizes the most influential work related to this research, to include the title of the work, modalities, key findings and other interesting information.

Table 2.1. A comprehensive list of the most related works and their contributions to this field of study.

Title of Work	Modality	Results	Other Information
Type Me the Truth!: Detecting Deceitful Users via Keystroke Dynamics [12]	Keyboard	Requiring no prior knowledge, associates typing behavior with production of lies regarding personal information	Supports cognitive load theory
Using Accelerometer and Gyroscope data in Common Mobile Devices to Assess Credibility [11]	Accelerometers and gyroscopes	Cheaters (liars) exhibited “greater values of acceleration/rotation rates than non-cheaters” [11]	Supports orienting response theory
Investigating the Effect of Insurance Fraud on Mouse Usage in Human-Computer Interactions [13]	Mouse movement	Deception increases normalized distance of movement, decreases speed of movement and increases response time	Consistent with the response activation model and supports deception theory and cognitive load theory
The Detection of Faked Identity Using Unexpected Questions and Choice Reaction Times [22]	Mouse and Keyboard Dynamics Recording	The most informative feature in distinguishing liars is the Inverse Efficiency Score, which combines speed and accuracy to unexpected questions	Using reaction times and unexpected questions, increased accuracy of liar prediction to 90% accuracy
Sleight of Hand: Identifying Concealed Information by Monitoring Mouse-Cursor Movements [3]	Mouse movement	Both electrodermal activity and mouse movement can be used to differentiate between truth tellers and liars	Captured both electrodermal activity and mouse analytics at the same time
Identifying Emotion by Keystroke Dynamics and Text Pattern Analysis [16]	Keystrokes	Analyzing keystroke timing attributes and text patterns resulted in above 80% accuracy of identifying emotions	Only seven emotional classes considered

2.3.1 HCI deception detection

Many studies have looked at how a computer user's behavior changes when acting deceptively. One study analyzed the ability to detect lies through keystroke dynamics utilizing randomized fields that indicated whether a participant should lie or tell the truth [12]. This study supported the cognitive load theory in that it showcased that people generally take longer to respond when lying than when telling the truth and that the number of edits (i.e. how many backspaces a person might use to correct incorrect typing) is positively correlated with deception [12]. In addition to typing generally faster responses when acting truthfully, one study also concluded that longer responses were typed when answering truthfully approximately 90% of the time [12]. On the other hand, the way that velocity measurements change between truth-tellers and liars is shown in Figure 2.1.

When trying to induce a response that mimics lying, it is imperative that the researchers create an environment where the given act is unsanctioned in order to ensure that the underlying psychological and physiological responses to said act (the lies) remain intact [23]. If a participant is aware that the nature of their deceit is without actual consequence, the underlying physiological responses will differ since there is no real risk of exposure, and therefore their senses may be less heightened. Indeed, producing completely authentic responses in a controlled environment can be difficult. A discussion of potential problems within an induced environment is provided in Chapter 6. Either way, induced deception should still deliver insights into how deception manifests in HCI.

Another study explored the phenomenon of the "orienting response" when users concealed information. This study utilized accelerometers and gyroscopes within hand-held electronics to prove that people respond to a triggering stimulus (i.e., being asked about a lie that they committed, with an orienting response, which is triggered by the cognitive load and the extra attention that they must give to the question) [11]. An orienting response can be a shifting of position, fidgeting, or another movement, either involuntary or as a reaction to the stimulus, and can be used to recognize deception in users. In these instances, the orienting responses were specific to individual users.

In a study about the effect of insurance fraud on mouse movement, the results suggest that lying produces an increase in the normalized distance of movement, a decrease in mouse movement speed, and an overall increase in the response time. The increased movement can

be attributed to cognitive competition between the truth and the lie, while the increase in response time can be attributed to cognitive load theory, and the increased processing to the time required to successfully commit the fraud [13]. Another notion is introduced in this work: the response activation model, which states that the brain begins to subconsciously program a response movement in order to fulfill the cognition's intention [13]. The response activation model indicates that if there is no conflict in a person's cognitive state, their mouse trajectory should follow a straight path towards the appropriate (ground truth) response. However, movement trajectories take a less direct path when distractions or ground truths are introduced, complicating the cognitive processes, and allowing the subconscious to temporarily deviate from the shortest trajectory based on the response activation model before the brain can fully commit to the lie [6].

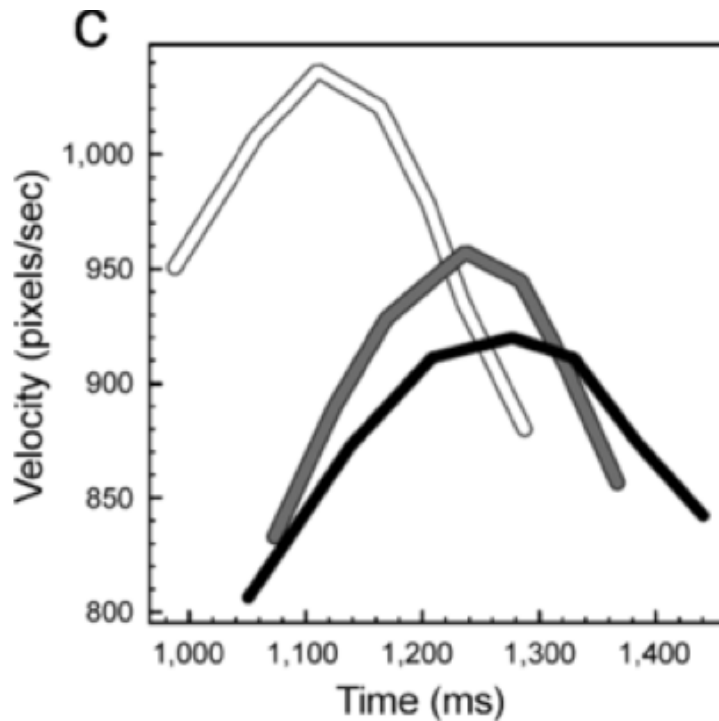


Figure 2.1. The white line represents a question with high truth values. The black line represents questions with the lowest truth values, and the grey line represents a medium or fluctuating truth value. The questions with the highest truth values witnessed the quickest mouse velocity responses, while the lowest truth values were the slowest. Source: [9].

While erratic or varied mouse movement had been explored in previous studies, Merylin Monaro and her colleagues explored the effects and differences of expected versus unexpected questions in the outcome and development of their responses [22]. In this study, an expected question refers to an auto-biographical question that could be found on one's identification card, such as a participant's middle name or date of birth. On the other hand, an unexpected question refers to a question that must be inferred from information on the identification card, such as a person's horoscope sign. Entering falsified information on an expected question could be as simple as copying or regurgitating information, but to answer an unexpected question, more critical thought must take place, and therefore, making it more difficult to falsify. In this study, the use of unexpected questions increased the accuracy of discriminating between truth-tellers and liars from 67% to 95% [22]. Monaro's works also looked at mouse trajectory, indicating that truthful user responses followed a more direct trajectory, while lying users followed a more deviated trajectory, as shown in Figure 2.2.

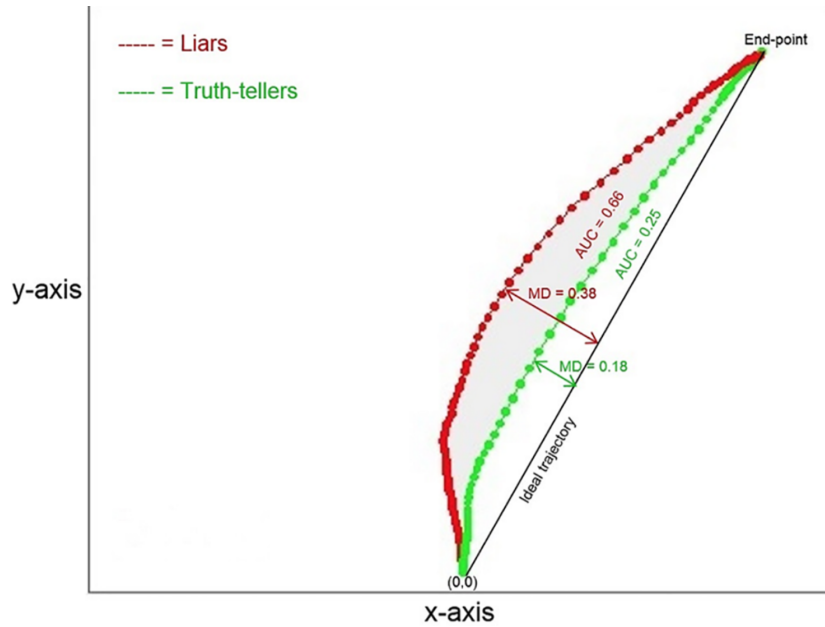


Figure 2.2. The black line indicates an ideal trajectory from the cursor's starting point to the location of the final answer. The green line represents the mouse trajectory of a truth-telling subject. The red line represents the mouse trajectory of a participant who is being deceptive. The deviation and upwards movement of the red line displays the subconscious competition between the two end-points in liars before finally committing to a lie, denoted as the "end point" in the upper right hand corner. Source: [22].

The use of HCI as a form of deception detection is already on-par with lie detection technology in use today. While both electro-dermal and mouse movements show potential for lie detection, mouse-cursor movements resulted in substantially fewer false positives than electro-dermal CITs techniques [3]. The difference in accuracy between the two approaches was about 15.6%, favoring mouse-cursor movement [3].

2.3.2 User-Specific Models

In general, online-based companies attempt to utilize technology to create a better user-experience or to increase profit and sales of their products. Such companies often rely on algorithms that run behind the scenes to create effective advertising schemes tailored to the specific individual. In essence, these companies utilize a user's online presence, routine, and patterns for targeted advertisements and marketing. This type of model is

one that is user-specific in nature, viz., it is tailored to each individual. These algorithms operate on the assumption of “normal human behavior,” and can experience problems when humans deviate from typical consumer behavior, which was seen throughout the coronavirus pandemic 2019 [4]. A study by Epp et al. examined how a user’s keystroke dynamics were altered depending on their current emotional state, which was self-reported at the time of keystroke capture [7]. The authors utilized fixed-text models as a baseline in order to analyze free-text models and how keyboard inputs shifted based on emotions such as confidence, hesitation, nervousness, relaxation, sadness, and even tiredness [7]. This study did not induce moods, but rather attempted to capture users’ interactions naturally throughout their days as they experienced a range of moods, which resulted in a lack of modeling for certain moods, such as anger, that are experienced less frequently. Additionally, this study relied on self-reporting in terms of mood analysis. Self-reporting can have potential downfalls, such as self-fulfilling prophecies and denial of certain feelings. For example, users who constantly feel anxious or depressed may have reported their feelings as relative, rather than absolute. Indeed, studies with self-reported emotional states have the potential for misleading correlation.

Past research that aims to harness components of HCI for deception detection have relied on similar user-specific models, which require historical analysis of individual users. While these models have tremendous benefits to computer scientists, our research aims to expand on these notions while eliminating the need for a user-specific model. Elimination of a user-specific model for HCI-based deception detection would expand detection usability and decrease insider threat response time. The ability to detect deception by a user who is unknown to a network has exponentially beneficial outcomes. The earlier an attempted deception is detected, the earlier the nefarious activity can be halted, examined, and corrected.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:

Survey Methodology

This chapter describes the survey creation, the methodology behind the survey, and the method of data collection from survey participants.

3.1 Experiment Overview

This work combines aspects of the studies detailed in Chapter 2 to find noticeable ways that deception manifests in human-computer interaction (HCI). Our study uses an online questionnaire to capture mouse movements, log keyboard interactions, as well as mouse clicks, which refer to the depression of a key on a computer mouse in order to either select or do something on the user's screen. To create a wide variation of expected responses, the survey uses a mix of expected and unexpected questions, baseline questions that are to be answered truthfully, and questions that are intended to induce deception. The details of the questionnaire follow. Through the use of the aforementioned question types, the questionnaire aimed to elicit natural, potentially deceptive, and qualitatively deceptive responses. *Naturally deceptive responses*, while unable to be measured or checked for legitimacy, are responses that capture a user lying in a natural setting. These *naturally deceptive responses* and potentially deceptive responses may occur when participants are asked sensitive questions that could potentially disparage their reputation, as users may wish to hide private or sensitive information from the researchers, such as their drug use or alcohol consumption. On the other hand, our questionnaire also utilizes *qualitatively deceptive responses* by asking users to answer a subset of questions while impersonating another person, thereby influencing the responses with some level of induced deception. While users are not necessarily lying in a natural state and their reactions and responses may vary, the presence of deception to some degree can be categorized and confirmed.

While previous works utilize user-specific models or binary choices (as detailed above when one answer appears on the left side of the screen and another on the right side), our research intends to include more input types in the model, without relying on user-specifics to create a model.

3.2 Questionnaire Format

3.2.1 Questionnaire Overview

The overall goal of the questionnaire was to create a variety of questions that would induce both natural, potentially deceptive, and qualitatively deceptive (pretend) responses and reactions from the participants. The questionnaire was split into two portions of identical length. For the first half of the survey, the participant is asked to answer questions truthfully. However, for the second portion of the survey, the participant is asked to impersonate a close friend, significant other, or parent, in order to imitate an insider threat. Having a user pose as another user about whom they have intimate knowledge provides insight into users whose aim is to impersonate others, falsify information on online forms, or even accidentally answer questions in a misleading manner. The questions that the participant is asked while they are impersonating an individual are nearly identical to those they were asked in the first portion of the survey. This provides consistency with respect to the questions being asked.

3.2.2 Question Response Types

The questionnaire employs a variety of submission formats, in order to capture various types of HCI throughout the survey. There are three main types of response formats: radio button response, drop-down menu selection, and text-box response. The submit button is uniformly placed on each page of the survey, centered and at the bottom of the screen. The placement of the button automatically adjusts to the participant's window size at the time of the survey. To promote consistency in collection, participants were asked to maximize their browser screen and complete the survey on a laptop or desktop computer.

Radio button responses are listed horizontally with equal spacing between each response. Questions with radio button responses are the primary form used to capture changes in trajectory based on deceitful intent. The radio buttons are mutually exclusive and only one answer may be submitted at a time. Participants are required to click the submit button to advance to the next question. Since the submit buttons are uniformly located on every page, each participant's cursor will be in the same location at the start of a new page. A consistent cursor start location is important so that the trajectory of mouse movement may be tracked throughout the question responses and study, shown in Figure 3.1, per a case study based on deception and mouse-cursor movements [3].

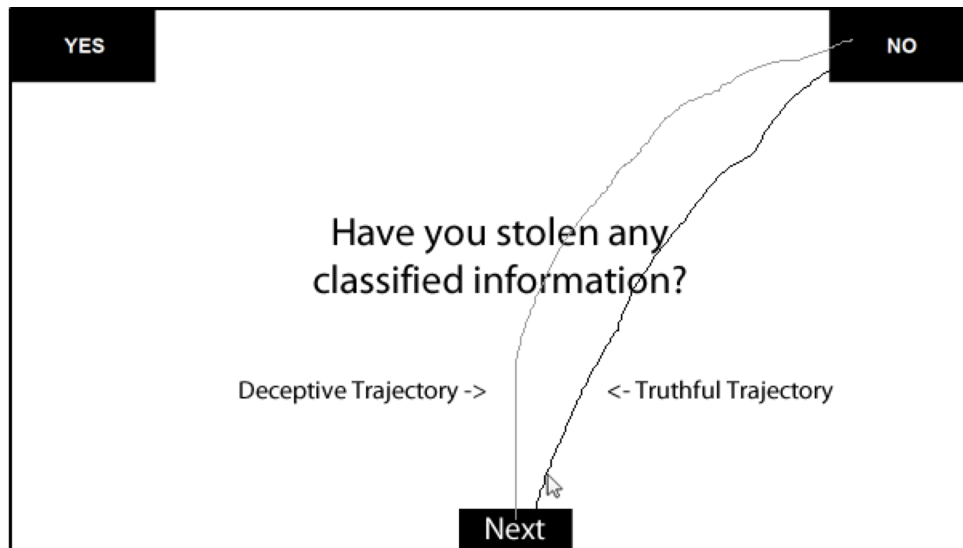


Figure 3.1. The black line on the right represents a truthful response. The (grey) line to the left is representative of hesitation that is attributed to potential deceit and the activation response model. In general, related studies have shown that mouse movement typically travels a more “direct path” when a user is telling the truth (highlighted by the dark black line). However, when a user is deceptive in nature, their cursor trajectory is less direct, and shows hesitancy in response (represented by the grey line that moves directly upward prior to moving towards the “no” on the image). Source: [3].

The next type of question submission mode is a drop-down menu selection. Drop down menu selections are used for questions that have too many responses to utilize the radio button submission, such as the month a participant was born.

The last type of question involved in the questionnaire is text response, which captures string inputs as the response method. The text response is the primary method of measuring keyboard interaction throughout the survey. Strings that are well-known to a user, such as their own first and last name, are expected to be typed more quickly and efficiently than information that may be more foreign to the user, such as another person’s first-owned car, or even an unfamiliar first and last name.

3.2.3 Categories of Questions

Our research subdivides questions into five hypothetical categories: white, green, yellow, red, and orange. Each of these categories corresponds to a different type of question designed to invoke a particular response from the survey participant. The motivation to include a variety of questions was drawn from various works, most prominently Merilyn Monaro's use of unexpected questions in [22], which explores how responses vary when a question is unexpected versus expected. Variation in mouse trajectory when introduced to an unexpected question versus an expected question is illustrated in Figure 3.2, from Monaro et al's research [22].

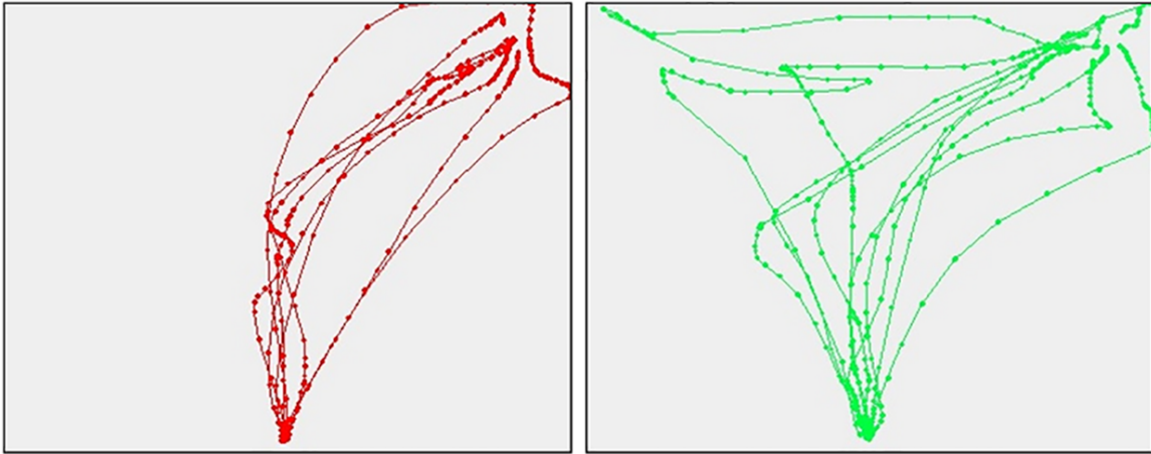


Figure 3.2. Both lines portray the trajectory of mouse movement during recorded responses from a controlled liar. On the left, the red lines are trajectories from expected questions. On the right, the green lines are trajectories from the same test subject when asked an unexpected question. Source: [22].

Our research expands upon this notion to create five separate categories that are expected to elicit different response mechanisms and response orientations. A breakdown of the five questions can be seen in Table 3.1.

Table 3.1. Categories of questions based on color-codes.

Question Type	Example	Participant Orientation	Intended Response
White	“What year is it?”	Truthful	Truthful and Accurate
Green	“What elementary school did you attend?”	Truthful	Truthful
Yellow	“Have you ever smoked marijuana before?”	Truthful	Varied
Red	“What is your first name?”	Deceptive	Deceptive
Orange	“What is your horoscope sign?”	Truthful	Unexpected

The white questions are intended to serve as a baseline for data collection. In this research, a white question is objective and can and should be answered at the time of survey by any and all participants in the same manner. Examples of white questions could be: “What year is it?” or “Who is the President of the United States?” These questions have an objectively correct answer, and since the survey was given throughout a consistent time window, the answers from all participants should be the same. In addition to objectivity, answers to white questions require only basic knowledge that every questionnaire candidate is expected to know. Answers from the baseline questions can be utilized to determine the participant’s commitment to the survey, as their answers can be easily compared to the known answers for correctness.

A green question in this study is a simple and expected question where the response elicited is intended to be truthful. While green questions are intended to elicit truthful responses, there is no mechanism in place to verify that a participant is telling a full truth, a partial truth, or no truth at all. Green questions were inspired and derived primarily from typical

security form questions regarding basic auto-biographical information and are similar to password reset questions. Green questions apply when the participant is asked to be truthful (acting on their own behalf) and a question is also expected – such as the elementary school they attended, or their first and last name. These questions are not intended to be difficult, and are intended to be questions that are highly trafficked when it comes to online form submissions.

A yellow question is a question where a participant is asked to be truthful, but where the researchers expect some deviation from the ground truth based on the heightened risk of the question, and the possibility of an orienting response. For example, a yellow question might ask whether or not a participant has ever smoked marijuana, or how many drinks a person consumes on a weekly basis. While participants may or may not answer truthfully, the sensitivity of such questions may heighten their level of awareness and correlate to how they orient their responses, which may be recognizable through their interaction with the human-computer interaction (HCI). Participants might exhibit a more noticeable orienting response to a yellow question because of the social consequences and implications of answering such a question truthfully. Because deception “requires heightened cognitive resources,” people tend to “exert more cognitive effort when concealing information” on key questions compared with answering benign baseline questions due to the sensitivity of the question and their desire to preserve their lie [3].

Red questions are questions where the survey participants are explicitly asked to provide deceptive information on the questionnaire. Red questions may take the same form as the green or yellow questions discussed above, but participants are expected to answer the question on behalf of another individual, which inherently makes the responses deceptive in nature. For these questions, the participants were asked to assume another identity, such as that of a significant other, close friend, or parent, and impersonate that person for the remainder of the survey. We asked participants to choose the person for whom they have the best chance of successful impersonation, in order to capture more legitimate insider threat data.

Orange questions are a subset of both red and yellow questions, but specifically refer to those questions that are both unexpected to the user and where the participant was asked to impersonate an individual. While an insider threat may showcase intimate knowledge of

basic biographical information, it may be more difficult to answer “unexpected” questions with the same ease [22]. In this case, an “unexpected” question refers to something that is tangentially related to basic biographical information that an insider threat may easily know, but asked in a manner that may not be as intuitive. For example, a horoscope sign is directly related to birth month and date. If someone knows a birth date, they can theoretically derive a horoscope sign if given the appropriate information or background knowledge. However, while it is quite natural to provide someone’s birth month as a string when given a birth date, it is not as natural, nor a typical security form question, to know the corresponding horoscope sign, which requires supplementary cognitive processes. In addition, many people may not even know their own horoscope signs and may lack this information altogether. While this “unexpected” question may provide insight for this study, it is possible that many people aiming to tell the truth simply cannot answer this question either. Another example of an unexpected question in relation to birth dates is knowing someone’s specific age. While it may be quite simple to write down a full birth date, and it is possible to calculate age from that date, it may not be as simple to quickly provide an associated age. This type of question introduces room for error and miscalculation that may be indicative of potential insider threats and deception that may be occurring.

This research aims to provide insight into certain insider threat activity that has not been studied in-depth yet. Our research primarily utilizes the impersonated form to attempt to understand certain insider threat behavior and activity. The intimate knowledge that one possesses within the interpersonal relationships that are being impersonated can be reflective of the extensive social engineering that takes place prior to high-profile and high-risk insider threat attacks on networks of interest, and therefore provides useful information when analyzing deception within HCI.

3.3 Survey Launch

3.3.1 Recruiting Survey Participants

All research participants were students or faculty at the NPS during the 2021 academic year. This research was approved by the Naval Postgraduate School (NPS) Institutional Review Board to ensure ethical research and practices. Participants were recruited from various academic programs and engaged in the experiment on a strictly voluntary basis. The first

e-mail invitation was sent to the entire computer science department at the NPS. To increase participation and diversity of responses, the next wave of e-mail invitations were sent out to non-computer-based programs, in an attempt to collect more varied data. Presumably, computer science and cyber operations students and faculty are more familiar with keyboards and mice than their counterparts in certain other academic programs. Including other programs was an attempt to include a wider range of computer competencies scattered throughout the collected data.

The survey link was embedded in the e-mail invitation and participants had the option to decline or participate in the survey by clicking on the secure link. When a user opted to participate in the study, their survey information was logged and stored in an individual comma-separated values (CSV) file, based on the web logging code discussed in Section 4.1. At any point, participants had the option to stop survey participation and exit the browser. If a participant no longer wished to partake in the survey, their incomplete survey was discarded and not included in data analysis. Survey participants were not rewarded in any way for their assistance and cooperation throughout the research process.

The email invitation asking students and faculty to participate in this study is shown in Figure 3.3.

Good morning!

You are invited to participate in a Naval Postgraduate School research study entitled “Zero-shot Insider Threat Detection through Human-computer Interaction.” The purpose of the study is to understand key components of insider threat activity based on how participants respond to some security questions. Your participation in this survey is strictly voluntary.

The survey consists of approximately 30 short questions and involves answering the first half of the questions truthfully as yourself, and, for the second half, in a deceitful manner as you attempt to impersonate someone you know very well.

Since security in the DoD is everyone’s responsibility, participation in this study will benefit the DoD as a whole, taking us a step closer to understanding insider threats.

To complete the survey, please click on the link below, or copy/paste the link into a web browser. We ask that you take the survey on a desktop or laptop computer with your browser window maximized. The survey should take you no longer than 5-10 minutes. Thank you for your time and please feel free to reach out with any questions/concerns you may have! Your consideration is much appreciated.

Survey Link

Very Respectfully,
LT Jamie Francona
Cyber Systems and Operations

Figure 3.3. The e-mail example that was sent to potential survey participants requesting participation in our study.

Once a participant clicked on the link provided in Figure 3.3, they were directed to the survey’s landing page, which provided a more thorough introduction to the associated research, survey completion instructions, as well as information regarding the storage of their responses and contact information for the Principal Investigator. The landing page is shown in Figure 3.4.

Deception detection survey

Thank for choosing to participate in the study entitled, "Zero-shot Insider Threat Detection through Human-computer Interaction." The purpose of the study is to understand key components of insider threat activity based on how participants respond to some security questions. Your participation in this survey is strictly voluntary.

Your responses to the following survey containing 32 questions will enable us to better understand and predict insider threats.

The survey is broken up into two parts. In the first part, you will be instructed to answer questions truthfully, or to the best of your knowledge.

In the second part, you will be instructed to impersonate someone you know well, answering questions as if you were that person. Try your hardest to avoid detection as an intruder!

All of your responses will remain anonymous and confidential. If at any time you wish to discontinue the questionnaire, please exit the browser. Incomplete questionnaires will not be included in research.

If you have any questions or comments about this study, please contact the Principal Investigator, Dr. Vinnie Monaco, (831) 656-7602, john.monaco@nps.edu. Questions about your rights as a research subject or any other concerns may be addressed to the Navy Postgraduate School IRB Chair, Dr. Larry Shattuck, 831-656-2473, lgshattu@nps.edu.

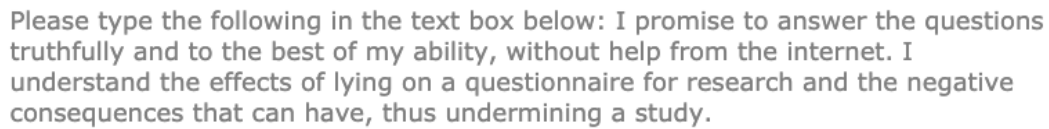
If you wish to start the survey, please click on the button below.



Figure 3.4. The page that a participant is directed to when clicking on the survey link. Of note, the "Begin" button is in the same location (bottom center of the page) as the "Submit" button that follows on subsequent pages to ensure that each participant shares a common starting point for mouse movement.

After consenting and continuing with the survey, each participant was asked to type a small paragraph into a text box. The requested paragraph text is detailed in Figure 3.5. The purpose of the paragraph text is two-fold. First, it enables the researchers to capture a baseline assessment of a participant's typing competency. In other words, this paragraph essentially serves as a pre-survey measurement of the participant's typing proficiency, use of the backspace and delete buttons, and to gather an overall assessment of the user's basic keyboard interactions. Second, the text serves to present an environment where lying is

sincerely disapproved, thereby elevating concern regarding possible punishment for being deceptive. In a way, the text prompt serves as an ad hoc “honor code,” which has been proven to “reduce people’s tendency to engage in fraudulent behavior,” and therefore heighten the participants’ orientation toward providing truthful responses [23]. With the implied request for ground truth, the response activation model may become clearer when participants intend to lie. It would create more hesitancy in response time as well as different mouse movements and keystroke rhythms, thus providing a more accurate depiction of how deceit manifests in HCI.

A screenshot of a text box containing the following text: "Please type the following in the text box below: I promise to answer the questions truthfully and to the best of my ability, without help from the internet. I understand the effects of lying on a questionnaire for research and the negative consequences that can have, thus undermining a study." The text is in a standard sans-serif font, with the first line in bold and the rest in regular weight. The background is a light gray.

Please type the following in the text box below: I promise to answer the questions truthfully and to the best of my ability, without help from the internet. I understand the effects of lying on a questionnaire for research and the negative consequences that can have, thus undermining a study.

Figure 3.5. The official start of the survey in which participants are asked to type out an honor code.

The complete list of survey questions can be found in Chapter 4, Section 4.2.3.

3.3.2 Responses to Survey

A number of students, particularly within the computer science and cyber operations curricula, expressed hesitation and concern with the administered survey. Practicing excellent “cyber hygiene,” these participants contacted members of the research team to ensure that the email and associated link were, in fact, legitimate and did not constitute an advanced phishing scheme. The majority of concerns with survey participation were based on the type of questions asked in the survey, which if handled incorrectly, could put participants’ vulnerable data at risk (i.e. answers to password reset questions). Out of the 107 participants who started the survey, only 32 completed it in its entirety and, therefore, could be included in the research. The relatively low completion rate of 29.9% can likely be attributed to personal concerns for privacy regarding their private information. While this response limited data collection to a degree, it gave the researchers confidence that the questions being asked would induce a legitimate activation response in participants.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:

Data Collection

Following our discussion of the survey in Chapter 3, this chapter continues with an explanation of how the data was collected through web-logging code. After, data pre-processing, manipulation, feature extraction and label creation are discussed.

4.1 Web Logging

To maintain participants' privacy, actual survey responses were not collected. Rather, the input events of interest (e.g., those related to HCI) were collected and stored for data manipulation.

Web logging code was created to capture the various input elements, such as mouse movement and keystrokes. First, a dictionary was created to map numerical codes to all of the different possible keyboard inputs, including the typewriter keypad, all of the function keys, system keys, cursor control keys, and enter keys.

To capture mouse events, the web logging code captures timestamps for various events to isolate mouse movement and mouse clicks. To gather information regarding trajectory, the code takes information about the X and Y coordinates of the mouse and calculates offsets using these coordinates.

Each keyboard and mouse event that took place during the online survey session is then logged with either correlating numeric codes or time data. These logged events are then loaded into a CSV file. Each survey response is automatically named by the time and date that the survey was taken.

4.2 Data Pre-processing

Data pre-processing includes data aggregation, data cleaning, as well as feature description and label creation.

4.2.1 Data Aggregation

Once the appropriate number of participant surveys was completed, the raw survey data collected in individual CSV files was aggregated into a single Pandas dataframe using a Python script. The raw data was then pre-processed for further exploitation and analysis. Pre-processing began with sorting complete surveys from incomplete surveys, dropping the incomplete surveys, and creating columns with basic calculations for movement and velocity statistics.

4.2.2 Features and Headers

Our web-logging code, previously discussed in Section 4.1, was programmed to capture data that, when loaded into a properly formatted dataframe, consisted of 10 columns: event, question, date_time, x, y, z, id, and filename. An example of the dataframe head is shown in Figure 4.1.

	event	question	time	date_time	performance_time	x	y	z	id	survey
0	mousemove	NaN	739.970	1617140387783	745.345	2659	606.0	0	content	survey_0
1	mousemove	NaN	757.250	1617140387799	761.325	2637	599.0	0	content	survey_0
2	mousemove	NaN	771.945	1617140387815	777.265	2613	594.0	0	content	survey_0
3	mousemove	NaN	787.910	1617140387832	793.955	2591	588.0	0	content	survey_0
4	mousemove	NaN	810.170	1617140387849	810.695	2570	584.0	0	content	survey_0

Figure 4.1. An example of the aggregated Pandas dataframe when the raw data is imported individual CSV files. This is the product of running `df.head()` on the aggregated dataframe. Data in this example was not included in the study.

Header Descriptions

Event: A categorical header, based on the web-logging code, to describe the discretely recorded events throughout the duration of the participant’s survey. Examples of events are: `mousemove`, `mousedown`, `mouseup`, `keydown`, `keyup`, `keypress`, and `mousewheel`.

Question: The question column associates the data in the respective row with the question ID from the survey’s JSON file. There are 33 unique question values recorded: `NaN` and

the numbers 0-31. The NaN values correlate to an empty question ID on the survey landing page. The numeric IDs correspond to the baseline paragraph response and the 31 survey questions that follow. The question header allows the researchers to keep track of the type of questions (see Section 3.2.3) in order to label and analyze HCI responses.

Date_time: The date_time column captures the calendar date, hour, minute, second and millisecond time data of the particular survey event in an int64 format. This event timestamp is obtained through the use of Date.now() in JavaScript.

X, Y, and Z: X and Y refer to the screen's plane and are utilized to capture mouse trajectory and position during the survey. X and y are both measured as screen offsets, in units of pixels. Z is a binary variable used to indicate pointer-dragging events. If a mouse button is held down during an event, Z is set to 1. Otherwise, Z is 0.

ID: The ID column labels the HTML element ID attribute for which the event occurred. Each radio-button and drop-down answer has an associated ID, which is logged when clicked or selected by a participant. For example, if a participant selects the submit button to advance to the next question, the data event would be "event: mousedown" with an associated ID of "submit." There are a total of 50 unique IDs. Tracking answer and submit IDs allows for further and more detailed analysis once the data is completely pre-processed.

Filename: When each individual survey is taken, a unique filename is generated based on the current filepath and then stored in the filename column.

Data Cleaning

As the CSV files were being read in from the shared folder into an aggregated dataframe, the number of unique questions were counted. If the number of unique values in the question column was less than 32, the survey file was dropped while being read-in. This action was based on the assumption of missing data or an incomplete survey. Partial surveys were discarded completely, and include surveys where people either skipped questions or did not take the survey in its entirety by stopping prematurely. Surveys with incomplete data were discarded. This ensured that only a set of consistent, complete data was analyzed. This reflects our choice to allow participants to withdraw or opt out of the survey at any time, as discussed in Section 3.3.1.

Once all appropriate files were aggregated into a dataframe, the raw data was cleaned and manipulated for analysis. First, we utilized a **groupby** operation to group the aggregated data by “survey” so that each participant’s survey responses would be grouped together. Once grouped properly, we sorted the values by *date_time* to ensure the proper order of events.

Next, we created three new columns including *delta_time*, *delta_position* and *velocity*. *delta_time* is a measurement of the change of time between recorded events while *delta_position* is a measurement of the change in position of the mouse movement based on the x and y coordinates for a particular event. The base assumption when *delta_time* is calculated is that $t_1 \leq t_2$. The *delta_position* is an absolute value calculation, so a positive or negative value does not denote a specific direction. *Velocity* refers to the instantaneous velocity of that discrete event, calculated from the *delta_position* and *delta_time* features. Since *delta_position* is calculated with absolute value, the *velocity* calculation is also directionless. The calculations for these new columns are as follows:

Listing 4.1: Change in position equation

```
delta_position = np.sqrt(df['x'].diff()**2 + df['y'].diff()**2)
```

$$\text{delta_position} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Listing 4.2: Change in time equation

```
delta_time = df.groupby('survey')['delta_time'].diff()
```

$$\text{delta_time} = t_2 - t_1$$

Listing 4.3: Velocity equation

```
velocity = df['delta_position']/df['delta_time']
```

$$\text{velocity} = \frac{\text{delta_position}}{\text{delta_time}}$$

Upon analysis of the newly calculated data, there were some velocities with a value of infinity (“inf” values in Python), pointing to erroneous calculations. A calculated “inf” value (representing infinity) based on the above calculation, is most likely due to a divide-by zero operation when calculating velocity. Therefore, under certain circumstances, the divisor in the velocity equation is zero, which refers to the change in time. For a change in time to be zero, the time of two sequential recorded events must take place within the same recorded time. Since the granularity of the clock is coarse throughout the web logging, duplicated timestamps can occur for different events, which then create erroneous velocities. Of note, the `Date.now()` which is utilized to capture the *Date_time* values has a 1ms resolution on Chrome, but a 2ms resolution on Firefox [24]. In some instances, these resolutions may be even coarser. To resolve this problem, we utilized the header *date_time* as a subset and any rows with the same exact *date_time* were dropped from the dataframe utilizing the following code:

```
df = df.drop_duplicates(subset=["date_time"])
```

4.2.3 Label Creation

Once the data was thoroughly cleaned, a final column was added to the dataframe, which labeled the data. Each row was given a label which corresponded to the question color categories discussed in 3.2.3. For example, if a question was a “baseline question,” it was given a label of “white.” Code was written that associated the question ID (NaN or any of 0-31) with its associated color category, as determined by the color codes created by the researchers in Section 3.2.3. Questions were labeled with the following unique color labels: Green, White, Yellow, Red, and Orange. In addition, questions with a label of “0.0” were categorized as “Baseline Text” and questions with “NaN” values were labeled as “Consent Page.” Description of the five basic color categories can be found in Section 3.2.3.

Complete List of Survey Questions

This section includes a comprehensive list of the survey questions asked throughout the questionnaire. The color of the text corresponds to the color-coded label for each respective

question. White questions are on a black background and yellow questions are on a purple background.

1. Baseline Text: "Please type the following in the text box below: I promise to answer the questions truthfully and to the best of the ability, without help from the internet. I understand the effects of lying on a questionnaire for research and the negative consequences that can have undermining a study."
2. What is your first name?
3. What is your middle name?
4. What elementary school did you attend?
5. What gender do you identify with?
6. What is the current year? (Example: 1997)
7. In what city/town were you born?
8. What was the make of the first car you owned?
9. What month were you born?
10. What is your horoscope sign?
11. What is your birth date? (Mon/Day/Year, example: 11/2/1997)
12. How old are you? (Example: 21)
13. How many alcoholic drinks do you have weekly?
14. Have you ever smoked marijuana?
15. What is the name of your first love?
16. Who do you trust more than anyone in this world?
17. "For the remaining set of questions, please pick a significant other, parent, or close friend to impersonate. You should choose the person whom you know the most information about and therefore are most likely to be able to answer the questions truthfully and accurately. Please specify from which category below your new identity belongs and answer the questions to the best of your ability.

Good luck!"

18. What is your first name?
19. What is your middle name?
20. What elementary school did you attend?
21. What gender do you identify with?
22. What country do you live in? (No abbreviations)
23. In what city/town were you born?
24. What was the make of the first car you owned?
25. What month were you born?
26. What is your horoscope sign?
27. What is your birth date? (Mon/Day/Year, example: 11/2/1997)
28. How old are you? (Example: 21)
29. How many alcoholic drinks do you have weekly?
30. Have you ever smoked marijuana?
31. What is the name of your first love?
32. Who do you trust more than anyone in this world?

A full description of the color categories can be found in Section 3.2.3.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:

Deception Detection Methodology

5.1 Deception Cues in HCI

Previous studies have looked at various forms of HCI, such as mouse trajectory, velocity of mouse movement, and even the speed at which one types, both as cues for deception and for other motivations [3], [12], [22], [25]. In most of these studies, user-specific models were utilized, comparing one user's movement cue during deception against that same user's movement cue during a truth-telling scenario, and creating a model based on the differences that specific user exhibited during the contrasting situations. In this study, our aim is to create a non user-specific model based on deception cues that could be applied to first-time users. This would allow their trustworthiness to be determined without the need for historical data and trend analysis on each user.

In this study, three main features were utilized for deception detection: *velocity*, *angle*, and *delta_time*.

5.1.1 Velocity as a Feature

Velocity, which was detailed in Chapter 4, Section 4.2.2, was calculated using the equation in Listing 5.1. In order to calculate *velocity*, which is the change in position over the change in time, *delta_position* was divided by *delta_time*, which were calculated previously using the equations in Listings 4.1 and 4.2 respectively.

Listing 5.1: Velocity equation

```
velocity = df['delta_position']/df['delta_time']
```

Dropping duplicated timestamp values and sorting by *date_time* as discussed in Section 4.2.2, in conjunction with using the **groupby** function in both the *delta_position* and *delta_time* calculations ensured that the *velocity* data was ordered by survey, and therefore, by user, and in the correct sequential order.

Velocity Histograms

We decided to utilize histograms as our feature data since it enabled us to capture various *velocity* profiles. These *velocity* profiles are discussed in more depth in Chapter 6 and many examples are seen in Section 6.3, which includes case studies of these profiles. Utilizing histogram features enabled more in-depth analysis regarding feature averages and standard deviations that shed light on user behavior later in the study.

Once the decision was made to use histograms for features, we needed to determine the appropriate bin edges that would be used in the Random Forests and Extra Trees Classifiers for the *velocity* feature. To do so, we plotted the *velocity* of all mousemove events in a histogram. The *velocity* histogram used to determine bin edges is shown in Figure 5.1.

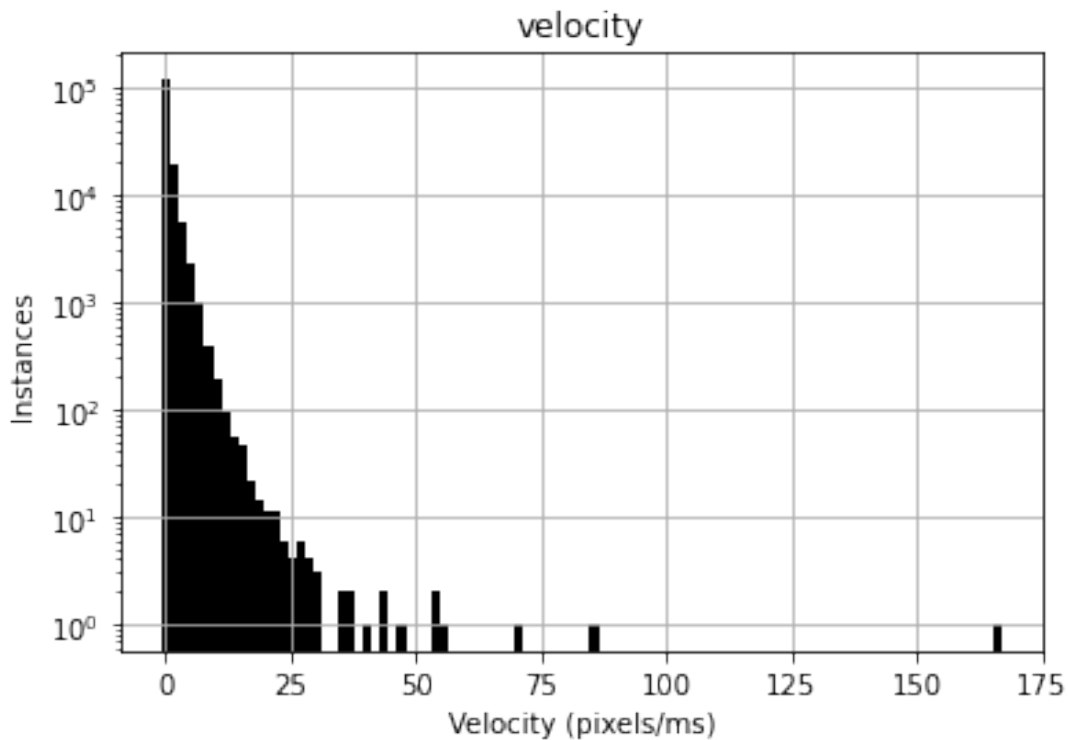


Figure 5.1. A histogram of all calculated velocities using 100 bins. This histogram represents a lower bin edge around 0 and an upper bin edge that begins to taper off around 30 pixels/ms.

The histogram in Figure 5.1 plots velocities that hover around zero and steadily decline until

approximately 27 pixels/ms, when the instances and density of velocities becomes increasingly sparse. In order to gain a deeper understanding of the tail end of the histogram, we ran quantile functions to determine the bottom and top 1% of velocities. When calculated, the lowest 1% of velocities was 0.0. On the other end, the top 1% of velocities was approximately 6.69 pixels/ms. The lower bin edge limit for *velocity* was always set throughout our experiment at 0.0 pixels/ms, while the upper bin edge was set to either a 10 or 20 pixels/ms, to account for some of the more rare activity seen on the tail end of the histogram in Figure 5.1.

5.1.2 Change in Time as a Feature

The next feature that we observed was *delta_time*, since it was previously calculated in order to determine *velocity*. *Delta_time* simply measures the change in time by taking the time differentials from one measured event to the next event. Our research included *delta_time* as a feature since it can capture a user's speed in performing certain actions, while avoiding some device-specific dependencies such as screen resolution and pointer acceleration, and can influence features like *velocity*.

Change in Time Histograms

In order to determine appropriate bin edges for the *delta_time* feature, we plotted the change in time for all mousemove events into a logarithmic histogram with 40 bins. The resulting histogram is shown in Figure 5.2.

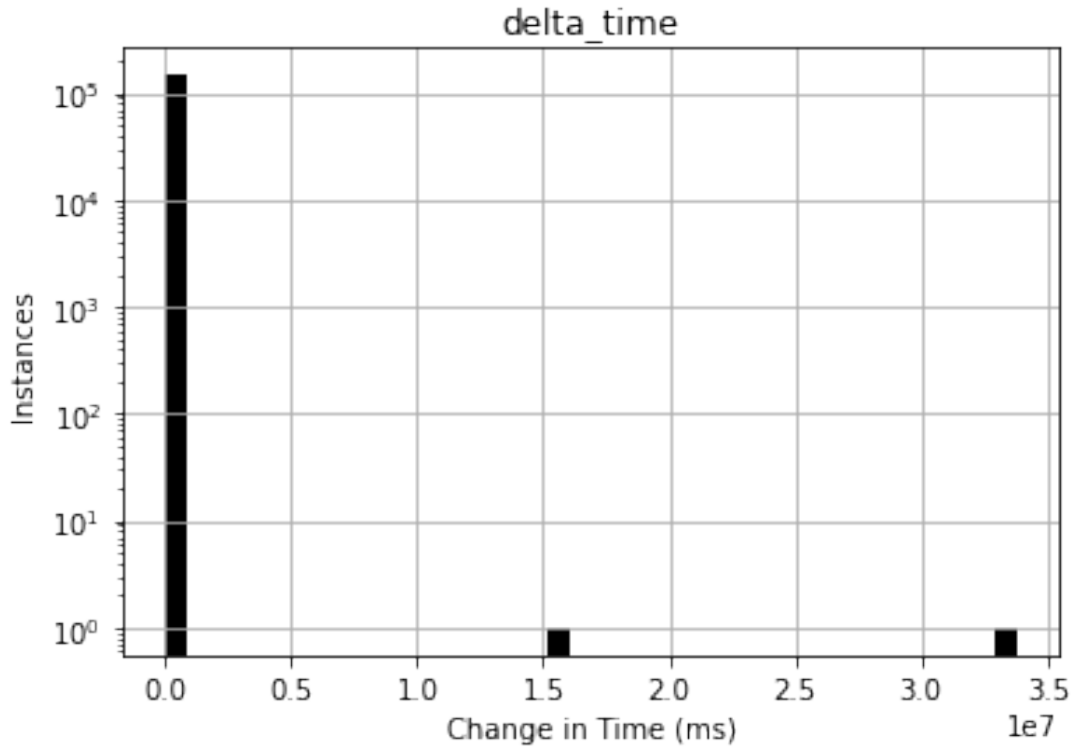


Figure 5.2. A histogram of all calculated changes in time using 40 bins. This histogram shows a lower bin edge of 0 and the majority of changes in time occurring between 0 and $0.2 \times 10^7 ms$.

Based on the histogram information, a lower bin edge of 0.0 ms was chosen, and an upper bin edge of 40,000 ms or 25,000 ms was utilized for *delta_time*.

5.1.3 Angle as a Feature

Lastly, we utilized *angle* as a feature to determine HCI cues. Unlike *velocity* and *delta_time*, which were previously calculated, *angle* needed to be calculated at this point in our analysis. In order to calculate the *angle*, we used the equation in Listing 5.2. The calculation takes the inverse tangent of the differential in x values squared and the differential in y values squared, thus calculating the hypotenuse of a triangle where the two legs are the change in y and the change in x. Once the hypotenuse of the triangle was calculated using the Pythagorean

Theorem, the inverse tangent is utilized to calculate the associated angle.

$$\theta = \arctan((y_2 - y_1)/(x_2 - x_1))$$

Listing 5.2: Angle equation

```
angle = arctan(df['x'].diff().values, df['y'].diff().values)
```

Angle Histograms

In order to determine appropriate bin edges for the *angle* feature, we plotted the angles for all mousemove events into a histogram utilizing 50 bins. The resulting histogram is shown in Figure 5.3.

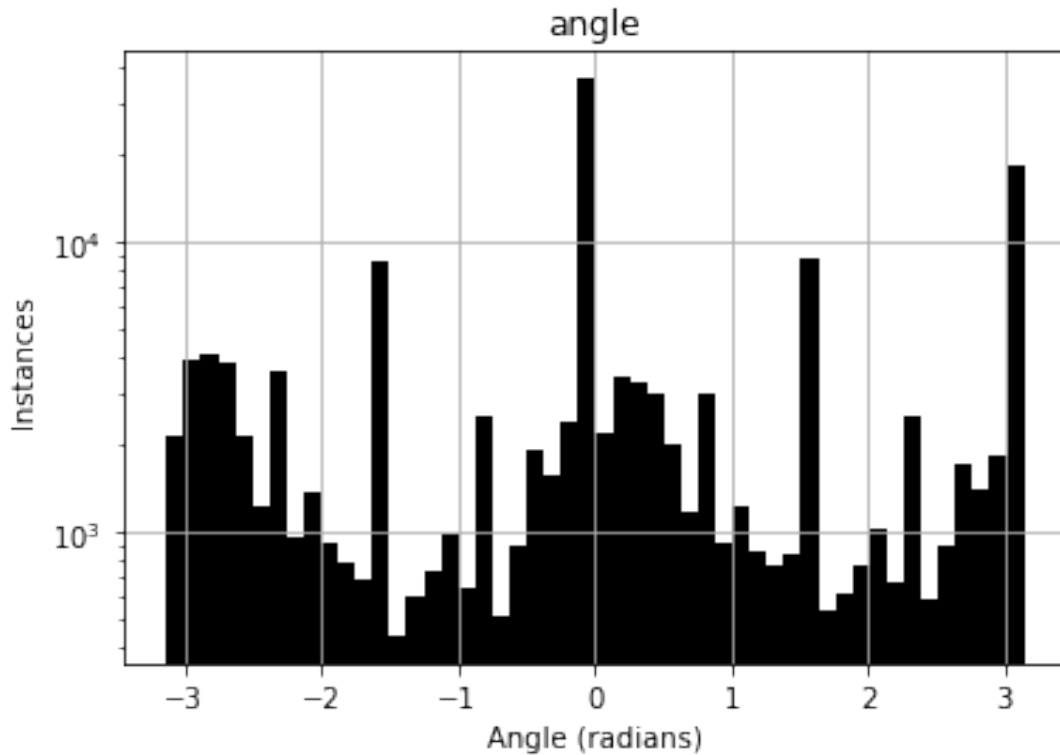


Figure 5.3. A histogram of all calculated angles using 50 bins.

Deducing information from the above figure, bin edges for *angle* were set to -3.0 radians

for the lower bin and +3 radians for the upper bin edge. The time interval over which the angle is measured corresponds to discrete mousemoves that the web logging code captures. Of note, there are four distinct modes that are present in Figure 5.3: -1.6 radians, 0 radians, 1.6 radians, and 3 radians. These four modes correlate with 0°, 90°, 180°, and 360°. These modes likely exist due to small mousemove captures that resulted in direction information only, vice full angle measurements.

5.2 Feature Extraction

A function was created to extract the features discussed in Section 5.1 from the overall dataframe. This function, called “extract_features” took in parameter x, some subset of the dataframe, as an argument and returned a Pandas Series containing histogram information of the associated feature values. The subset of the dataframes that was run through “extract_features” can be found in Table 5.1. If either the entire dataframe or a portion of it was used as a parameter in the “extract_features” function, each dataframe utilized the **groupby** function to appropriately group by the survey (user), question, and the question label (color). An example of the “extract_features” function and its application are detailed in Figures 5.4 and 5.5.

Extract Features Function

```
#create "extract_features" function for VELOCITY, ANGLE and DELTA_TIME
bins = 'auto'
def extract_features(x):
    velocity_hist = np.histogram(x['velocity'].values, density=True, bins = np.linspace(0, 10, 51))[0]
    angle_hist = np.histogram(x['angle'].values, density=True, bins = np.linspace(0, 1.6, 101))[0]
    time_hist = np.histogram(x['delta_time'].values, density=True, bins = np.linspace(0, 40000, 41))[0]
    return pd.Series(np.r_[velocity_hist, angle_hist, time_hist])
```

Figure 5.4. An example of the extract_features() function that uses all three features.

In Figure 5.4, histogram information of *velocity*, *angle*, and *delta_time* are all stored and returned in the Pandas Series. Each histogram that is created sets the density parameter to True, thus converting a normal histogram into a density plot. The density plot converts values on the histogram into probability densities [26], rather than discrete values. The bins parameter contains three elements of information. The first and second values of the bin

parameter correspond to the lower and upper bin edges, as specified in Sections 5.1.1, 5.1.2, and 5.1.3 respectively. The third value of the bin parameter corresponds to one more than the number of bins to be used. For example, if the number of bins is set to 51, this corresponds to 50 bins. Python’s numpy (imported in our Jupyter notebook as np) **linspace** method is utilized to create the number of bins, evenly spaced by appropriate size, throughout the histogram. The entire histogram is indexed by the first element in order to extract a series that can be used to fit the model.

Extract Features Application

```
#RUN FOR WHOLE DF
features = df.groupby(['survey', 'question', 'question_label']).apply(extract_features)
groupby_survey_question = df.groupby(['survey', 'question'])
labels_sorted = groupby_survey_question['question_label'].apply(np.array)
```

Figure 5.5. The code utilized to run the `extract_features()` function on the entire dataframe.

In Figure 5.5, a **groupby** function is applied to the entire dataframe that sorts the data by survey, question and question_label. Once the dataframe is grouped, it is passed to the “extract_features” function and the resulting histogram that is returned is saved in a variable called *features*. The *features* variable is an array of probability densities that make up the probability density histogram calculated in the “extract_features” function. Next, a labels vector was created to store the respective question_labels discussed in Section 3.2.3, such as red, green, or yellow, in an array format. The labels vector was calculated by utilizing the following line of code:

```
y = features.index.get_level_values('question_label').values
```

5.3 Models

In this study, both the Random Forest Classifier and the Extra Trees Classifier were used. With 32 complete surveys, a Random Forest algorithm allowed us to utilize a “bagged decision tree model,” allowing for minimization of variance [27]. In addition, Random Forest and Extra Trees both have quick training times, which allowed us to train and

run the model through many iterations without a large computational cost. To prevent, or attempt to prevent over-fitting the model, close attention was paid to tuning the model's hyperparameters, such as adjusting the number of `n_estimators` from a default of 100 to various sizes.

5.3.1 Model Validation

Once the appropriate features and label vectors were created, per Section 5.2, the data was split into training and test sets through the use of the **scikit-learn** library **`train_test_split`** function [28]. **`Train_test_split`** takes in parameters and splits them along a specified percentage into training and test sets. In our **`train_test_split`**, we passed in the feature vector (X), the labels vector (y) and the associated survey_id (s_id). A specified test_size of 0.3 was indicated, which created a test set equal to 30% of the total data passed into the **`train_test_split`**, leaving 70% of data in the training set. In addition, the **`stratify`** parameter was set to y, which helps to ensure distribution of data between the train and test set to ensure more accurate results. Utilizing stratify enables more even distribution of data between these two sets, which allows for more representative samples throughout the **`train_test_split`**.

5.3.2 Model Fitting, Training, and Accuracy

The model, either utilizing a Random Forest Classifier or an Extra Trees Classifier, was then fit using the imported classifier from `sklearn.ensemble` and passing in the associated `X_train` and `y_train` information that was calculated using the **`train_test_split`** discussed in Section 5.3.1. Once the model was fit, the accuracy of the model was the calculated by comparing the `X_test` to the `y_test`, or rather, the feature prediction of the test set to the actual labels of the test set. Lastly, the baseline accuracy was calculated, which represented the chance accuracy for the given iteration based on the data being modeled. The equation in Listing 5.3 was used to calculate baseline accuracy. Once baseline accuracy was calculated, the accuracy of the model could be compared to the baseline accuracy to determine the relative increase or decrease in accuracy that the model provided.

Listing 5.3: Baseline accuracy equation

```
baseline_accuracy = df[ 'question_label' ].value_counts().max()/len(df)
```

5.4 Splitting Data into Subsets

To facilitate various iterations of modeling, subsets of the data were created to separate the data both by question types and by question colors. The dataframe was originally segregated based on question type, and then further separated by color pairs, in order to create a binary classification problem, such as red versus green questions, for example. A comprehensive list of the data subsets can be found in Table 5.1. Each of these subsets is used to fit models used for prediction in order to determine which HCI deception cues manifest throughout the various question types.

The results from the modeling are displayed and discussed in Chapter 6.

Name of Subset in Code	Question Types Included	Binary Classification Colors
df	The entire dataframe, including text-based, radio-button, selection, and checkbox questions	Green + Red
df	The entire dataframe, including text-based, radio-button, selection, and checkbox questions	Green + Yellow
df	The entire dataframe, including text-based, radio-button, selection, and checkbox questions	Green + Orange
df	The entire dataframe, including text-based, radio-button, selection, and checkbox questions	Green + White
df	The entire dataframe, including text-based, radio-button, selection, and checkbox questions	Red + Orange
df	The entire dataframe, including text-based, radio-button, selection, and checkbox questions	Yellow + Red
df	The entire dataframe, including text-based, radio-button, selection, and checkbox questions	Yellow + Orange
df_radio	Only radio-button questions	Green + Red
df_radio	Only radio-button questions	Green + Orange
df_text	Only text-based questions	Green + Red
df_text	Only text-based questions	Green + Orange

Table 5.1. A comprehensive list of data subsets used throughout modeling and training.

CHAPTER 6:

Results and Discussion

This chapter explores the results of various model iterations and discusses the further exploration of certain results based on the associated prediction accuracy. After sharing the results of the study, the chapter includes a discussion regarding the overall study, its contributions, and areas of possible improvement.

6.1 Model Accuracy and Results Using a Random Forest Classifier

Utilizing the cleaned dataset discussed in Section 4.2.2 and the model parameters chosen in Chapter 5, the following accuracy results were obtained for the specified iterations run using the Random Forest Classifier. Unless otherwise specified throughout the text or in the associated tables, all modeling was based upon an ensemble containing 100 members.

The baseline accuracy calculated per the equation in Listing 5.3 was used for comparison of accuracy results produced by each model. Baseline accuracy in this case refers to chance accuracy of randomly choosing the correct binary classification. In order to determine relative increases or decreases in accuracy that each model created, the baseline accuracy for that iteration was compared to the accuracy produced by the classifier.

6.1.1 Velocity as a Feature

To determine a starting point, we ran single-feature binary classifications of all possible combinations of question colors utilizing *velocity* as the sole feature. The entire table of results can be found in Table 6.1. Overall, *velocity* did *not* show an increase in accuracy when classifying all green versus red questions, but did show increased accuracy potential, though marginal, for every other combination of question colors. The largest percent increase in accuracy was seen when classifying yellow versus red questions, which increased 12.92% from its baseline accuracy of 64.95% to 77.87%. This classification distinguishes between unexpected questions given to a truthful participant (yellow) and expected questions given to an impersonator (red).

Test	Bin Edges	# Bins	Acc.	Baseline Acc.	Δ Acc.
Green vs. Red	0,10	100	58.73%	59.87%	-1.14%
Green vs. Red	0,20	50	56.79%	59.87%	-3.08%
Green vs. Yellow	0,10	100	80.14%	73.44%	+6.7%
Green vs. Yellow	0,20	50	75.17%	73.44%	+1.73%
Green vs. Orange	0,10	100	73.33%	72.20%	+1.13%
Green vs. Orange	0,20	50	74.67%	72.2%	+2.47%
Green vs. White	0,10	100	85.61%	85.26%	+0.35%
Green vs. White	0,20	50	85.61%	85.26%	+0.35%
Red vs. Orange	0,10	100	73.08%	63.51%	+9.57%
Red vs. Orange	0,20	50	72.30%	63.51%	+8.79%
Yellow vs. Red	0,10	100	77.87%	64.95%	+12.92%
Yellow vs. Red	0,20	50	75.41%	64.95%	+10.46%
Yellow vs. Orange	0,10	100	46.96%	51.56%	-4.6%
Yellow vs. Orange	0,20	50	53.03%	51.56%	+1.47%

Table 6.1. Breakdown of accuracy results using Random Forest machine learning on binary classifications using velocity as the feature.

Rather than utilizing the entire dataframe, and therefore every question type included in the survey, the next set of simulations isolated just radio button questions and re-ran the binary classifications on red versus green questions and green versus orange questions only. When isolated, radio button questions showed a 16.36% increase in accuracy when classifying red versus green questions using only *velocity* as a feature, compared to a decrease in change in accuracy when all question types were included, as seen in Table 6.1. This indicates a 16.36% increase in accuracy when distinguishing participants that are telling the truth versus participants who are deceiving in some manner on expected, auto-biographical questions. Additionally, the largest increase experienced when classifying green versus orange radio button questions was 8.0%, a 5.53% increase in accuracy of the same classification when question type was not isolated. Full results of using *velocity* as the sole feature on radio button questions can be found in Table 6.2.

Test	Bin Edges	# Bins	Acc.	Baseline Acc.	Δ Acc.
Green vs. Red	0,10	100	47.37%	52.06%	-4.69%
Green vs. Red	0,20	50	63.16%	52.06%	+11.1%
Green vs. Red	0,10	50	68.42%	52.06%	+16.36%
Green vs. Orange	0,10	100	63.16%	55.13%	+8.0%
Green vs. Orange	0,20	50	36.84%	55.13%	-18.29%
Green vs. Orange	0,10	50	57.89%	55.13%	+2.76%

Table 6.2. Breakdown of accuracy results using Random Forest machine learning on binary classifications of radio button questions only using velocity as the feature.

Not surprisingly, the accuracy results when the text-based questions were separated and run with *velocity* as the sole feature were sub-par. Because *velocity* as a feature focuses on the rate of change in position as a function of time, it is less prone to capturing elements of typing, which are used for text-based responses, than mouse movements, which are used for other types of multiple choice responses. The change in accuracy of these iterations remained within 4.22% of baseline accuracy. This limited change in accuracy in either direction denotes unimproved modeling for this type of question, as the accuracy remains within a margin of error representative of guessing. Full results of using *velocity* only on text-based questions are detailed in Table 6.3.

Test	Bin Edges	# Bins	Acc.	Baseline Acc.	Δ Acc.
Green vs. Red	0,10	100	51.57%	55.79%	-4.22%
Green vs. Red	0,10	200	52.20%	55.79%	-3.59%
Green vs. Red	0,10	50	56.60%	55.79%	+0.81%
Green vs. Orange	0,10	100	90.42%	90.85%	-0.43%
Green vs. Orange	0,20	50	90.42%	90.85%	-0.43%
Green vs. Orange	0,10	50	91.49%	90.85%	+0.64%

Table 6.3. Breakdown of accuracy results using Random Forest machine learning on binary classifications of text-based questions only using velocity as the feature.

6.1.2 Angle as a Feature

The next individual feature that was explored was *angle*. When applied to the entire dataframe of all question types, *angle* indicated more potential when distinguishing red versus green questions compared to *velocity*, which showed no positive increase from baseline accuracy. *Angle* was able to increase baseline accuracy by approximately 16.36%. Additionally, *angle* produced a much greater increase in accuracy for the yellow versus orange classification compared to *velocity*. Although *angle* improved the green versus red and yellow versus orange classifications, it performed worse when classifying all other binary classifications. The full results of utilizing *angle* as the sole feature on the entire dataset are detailed in Table 6.4.

Test	Bin Edges	# Bins	Acc.	Baseline Acc.	Δ Acc.
Green vs. Red	-3,3	50	57.89%	59.87%	-1.98%
Green vs. Yellow	0,1.6	50	73.68%	73.44%	+0.24%
Green vs. Orange	0,1.6	50	73.68%	72.20%	+1.38%
Green vs. White	0,1.6	50	73.68%	85.26%	-11.58%
Red vs. Orange	0,1.6	50	63.16%	63.51%	-0.35%
Yellow vs. Red	0,1.6	50	63.16%	64.95%	-1.79%
Yellow vs. Orange	0,1.6	50	68.42%	51.56%	+16.86%

Table 6.4. Breakdown of accuracy results using Random Forest machine learning on binary classifications using angle as feature on entire dataframe.

When running the Random Forest Classifier with radio button questions only using *angle* as the primary feature, both green versus red and green versus orange classifications increased in accuracy. Green versus orange classification accuracy only increased by 2.76%, producing less favorable results when compared to *velocity* as the sole feature. However, the green versus red classification increased in accuracy by 21.62%. This result was 5.26% higher in accuracy than that achieved by *velocity* alone. The full data table displaying results of *angle* on radio button questions is detailed in Table 6.5.

No modeling was performed on text-based questions only using *angle*, since angle is not a relevant feature for keyboard input. Additionally, selection and checkbox questions performed poorly when in isolation, and thus, we focused primarily on distinguishing green versus red questions within the entire dataset and then distinguishing green versus red questions that utilized radio buttons in isolation for the remainder of the study.

Test	Bin Edges	# Bins	Acc.	Baseline Acc.	Δ Acc.
Green vs. Red	-3,3	50	73.68%	52.06%	+21.62%
Green vs. Red	-3,3	50	63.16%	52.06%	+11.10%
Green vs. Orange	-3,3	50	57.89%	55.13%	+2.76%

Table 6.5. Breakdown of accuracy results using Random Forest machine learning on binary classifications of radio button questions only using angle as feature.

6.1.3 Delta Time as a Feature

The last feature to be examined individually was *delta_time*. Though both *velocity* and *angle*, when utilized as individual features, had exhibited an increase in accuracy for every classification besides green versus red, *delta_time* actually illustrated a decrease in accuracy for both green versus red and green versus white. All other changes in accuracy were similar to those indicated in Table 6.1 and Table 6.4. All results based on the effect of *delta_time* on the entire dataframe are presented in Table 6.6.

Test	Bin Edges	# Bins	Acc.	Baseline Acc.	Δ Acc.
Green vs. Red	0,40000	50	58.25%	59.87%	-1.62%
Green vs. Yellow	0,40000	50	80.85%	73.44%	+7.41%
Green vs. Yellow	0,40000	100	82.27%	73.44%	+8.83%
Green vs. Orange	0,40000	50	78.00%	72.20%	+6.80%
Green vs. Orange	0,40000	100	73.33%	72.20%	+1.13%
Green vs. White	0,40000	50	79.55%	85.26%	-5.71%
Green vs. White	0,40000	100	83.33%	85.26%	-1.93%
Red vs. Orange	0,40000	50	70.00%	63.51%	+6.49%
Red vs. Orange	0,40000	100	65.38%	63.51%	+1.87%
Yellow vs. Red	0,40000	50	77.87%	64.95%	+12.92%
Yellow vs. Red	0,40000	100	76.23%	64.95%	+11.28%
Yellow vs. Orange	0,40000	50	45.45%	51.56%	-6.11%
Yellow vs. Orange	0,40000	100	57.57%	51.56%	+6.01%

Table 6.6. Breakdown of accuracy results using Random Forest machine learning on binary classifications using change in time on entire dataframe.

Once again, when analyzing radio buttons in isolation with *delta_time* as the sole feature, green versus red classification exhibited an increase in accuracy, rather than a decrease in accuracy as was the case when the entire dataframe of questions was included. Green versus red classification displayed a 21.62% increase in accuracy when using *delta_time* as the only feature, increasing the baseline accuracy of 52.06% to a model accuracy of 73.68%. *Delta_time* created the highest increase in accuracy for any single feature used throughout modeling of green versus red classification. The next highest-achieving feature was *velocity*, which accrued only 5.26% less accuracy during modeling than *delta_time*. On the other hand, this was the only single feature that actually decreased the prediction accuracy of the green versus orange classification on radio button questions in isolation. Table 6.7 depicts the results of modeling with *delta_time* as the sole feature on radio button questions.

Test	Bin Edges	# Bins	Acc.	Baseline Acc.	Δ Acc.
Green vs. Red	0,40000	40	73.68%	52.06%	+21.62%
Green vs. Red	0,40000	100	73.68%	52.06%	+21.62%
Green vs. Orange	0,40000	50	42.11%	55.13%	-13.02%

Table 6.7. Breakdown of accuracy results using Random Forest machine learning on binary classifications of radio button questions only using delta time as feature.

6.1.4 Velocity and Angle as Combined Features

After each feature was modeled individually, combinations of the features were explored. First, *velocity* and *angle* were combined for deeper model exploration. Accuracy increase of both binary classifications was higher when utilizing *angle* and *velocity* in combination, compared to using each feature individually. Using this combination for modeling, the baseline accuracy was increased from 52.06% for green versus red classification to 78.95% accuracy, a total increase of 26.89%. The change in accuracy for green versus orange classification mirrored the results produced by *velocity* alone in Table 6.2, increasing accuracy by about 8.00%. Results of using *velocity* and *angle* on radio button questions only are detailed in Table 6.8.

Test	Vel. Edges/Bins	Angle Edges/Bins	Acc.	Baseline	Δ Acc.
Green vs. Red	0,10,100	-3,3,50	57.89%	52.06%	+5.29%
Green vs. Red	0,10,100	0,10,100	78.95%	52.06%	+26.89%
Green vs. Orange	0,10,50	-3,3,50	63.16%	55.13%	+8.03%

Table 6.8. Breakdown of accuracy results using Random Forest machine learning on binary classifications of radio button questions using angle and velocity in combination as feature.

6.1.5 Velocity and Delta Time as Combined Features

The last pair of features used together in the model were *velocity* and *delta_time*. The increase in accuracy from baseline during green versus red classification was 21.62%, the same increase that was caused by using *delta_time* alone. All results, including bin edges, bins, and n-estimators, are detailed in Table 6.9.

Test	Velocity	Time	N-Estimators	Acc.	Baseline	ΔAcc.
Green vs. Red	0,10,50	0,40000,40	220	68.42%	52.06%	+16.36%
Green vs. Red	0,10,50	0,40000,40	100	63.15%	52.06%	+11.09%
Green vs. Red	0,10,50	0,40000,40	200	68.42%	52.06%	+16.36%
Green vs. Red	0,10,50	0,40000,40	300	73.68%	52.06%	+21.62%
Green vs. Red	0,10,50	0,25000,50	100	73.68%	52.06%	+21.62%

Table 6.9. Breakdown of accuracy results using Random Forest machine learning on binary classifications of radio button questions only using velocity and delta time as combined features.

6.1.6 Combining Velocity, Angle, and Delta Time as Features

When utilizing all three features on just radio button questions, accuracy increased by 16.36% for green versus red questions and by 13.29% for green versus orange questions. The accuracy increase for green versus red was similar to that produced by using *velocity* alone. However, the increase in accuracy using all three features was less than that produced by *angle*, *delta_time*, and the combination of *angle* and *velocity*. While green versus red did not display improved results, the classification of green versus orange was improved, scoring approximately 5.29% higher accuracy than any other combination produced. The breakdown of bin edges, bins and accuracy scored can be found in Table 6.10.

Test	Velocity	Angle	Δ Time	Acc.	Baseline	Δ Acc.
Green vs. Red	0,10,50	-3,3,50	0,40000,40	68.42%	52.06%	+16.36%
Green vs. Orange	0,10,100	0,2,100	0,40000,100	68.42%	55.13%	+13.29%

Table 6.10. Breakdown of accuracy results using Random Forest machine learning on binary classifications of radio button questions only using delta time, velocity and angle as features in combination.

6.2 Model Accuracy and Results Using the Extra Trees Classifier

Next, we utilized the Extra Trees Classifier, rather than the Random Forest Classifier. We first ran the model with all red and green questions in the entire dataset. The results are shown in Table 6.11. The modeling done in Figure 6.11 is used to develop the per-user accuracy results depicted in Figure 6.2 and graphically in Figure 6.2.

Test	Velocity	Angle	Δ Time	Acc.	Baseline	Δ Acc.
Green vs. Red	0,10,50	-3,3,50	0,40000,40	59.22%	59.87%	-0.65%

Table 6.11. Breakdown of accuracy results using Extra Trees Classifier on binary classification of all red versus green questions using velocity, delta time and angle as features in combination.

Next, we utilized the Extra Trees Classifier on radio button questions only using all three features in conjunction. The results of this model are in Table 6.12. Overall, accuracy increased by approximately 13.81%, a considerable increase, but still less than the model trained by the Random Forest Classifier produced.

Test	Velocity	Angle	Δ Time	Acc.	Baseline	Δ Acc.
Green vs. Red	0,10,50	-3,3,50	0,40000,40	78.94%	59.87%	+13.81%

Table 6.12. Breakdown of accuracy results using Extra Trees Classifier on binary classification of all red versus green questions using velocity, delta time and angle as features in combination.

Next, we trained an Extra Trees classifier with select features, *velocity* and *delta_time*, hoping to improve upon the accuracy results demonstrated with the Random Forest Classifier in Section 6.1. A model that was trained with the Extra Trees Classifier using *velocity* and *delta_time* as the features performed the best when classifying green versus red questions, increasing baseline accuracy of 52.06% by a total of 37.41%, bringing the model’s prediction accuracy to the highest seen in the study, 89.47%. The results of using Extra Tress on a variety of bin edges is detailed in Table 6.13. This combination of features, utilizing the Extra Trees Classifier, significantly outperformed any other combination of features and the Random Forest Classifier altogether.

Test	Velocity	Time	Acc.	Baseline	Δ Acc.
Green vs. Red	0,10,50	0,40000,40	89.47%	52.06%	+37.41%
Green vs. Red	0,10,100	0,40000,40	78.95%	52.06%	+26.89%
Green vs. Red	0,10,50	0,40000,50	84.21%	52.06%	+32.15%

Table 6.13. Breakdown of accuracy results using Extra Trees machine learning on binary classifications of radio button questions only using velocity and time as features.

6.3 Prediction Examples

Breaking the model’s predictions down by survey, and therefore isolating individual users, allowed us to analyze per-user accuracy to determine how the model affected individual users. We first looked at per-user accuracy when using the entire dataframe to classify red versus green questions. This classification included all question types, such as radio button,

selection, checkbox and text questions. While the overall model accuracy actually decreased from the baseline accuracy of 59.87% to a model accuracy of 59.22%, looking at per-user accuracy provides insights on how non user-specific modeling can vary from sample to sample.

Figure 6.1 provides the per-user accuracy results of each survey. Since the entire dataframe of red and green questions was included in the modeling, every single survey was included in the test set, and therefore, a per-user accuracy of each survey could be calculated. The entire survey consisted of 12 green and 10 red questions, creating a total of 22 questions to be considered for this prediction. Out of the 32 surveys, 19 surveys scored a higher accuracy than the model prediction (indicated by the green bars in Figure 6.2, while the remaining 13 surveys scored lower than the model accuracy (indicated by the red bars in Figure 6.2. Of note, out of the 13 users whose accuracy was lower than model accuracy, four users had an accuracy within 5% of model accuracy. The overall standard deviation for per-user accuracy was 21.65%. The highest user accuracy was 100%, occurring once for survey 1. This perfect accuracy score is 40.78% higher than model accuracy. The lowest recorded accuracy was 0% by survey 29, accounting for a 59.22% decrease in accuracy from model accuracy. Overall, 25% of participants were classified at least 16% more accurately using the Extra Trees Classifier than using baseline accuracy. Out of the 32 surveys, 7 were within 5% of 59.22% accuracy, representing a 21.88% cluster around model accuracy.

```
In [3354]: user_accuracy
```

survey_0	0.750000
survey_1	1.000000
survey_10	0.833333
survey_11	0.333333
survey_12	0.900000
survey_13	0.416667
survey_14	0.666667
survey_15	0.400000
survey_16	0.571429
survey_17	0.625000
survey_18	0.750000
survey_19	0.666667
survey_2	0.666667
survey_20	0.428571
survey_21	0.800000
survey_22	0.666667
survey_23	0.571429
survey_24	0.625000
survey_25	0.625000
survey_26	0.571429
survey_27	0.200000
survey_28	0.200000
survey_29	0.000000
survey_3	0.500000
survey_30	0.555556
survey_31	0.600000
survey_4	0.333333
survey_5	0.666667
survey_6	0.666667
survey_7	0.800000
survey_8	0.833333
survey_9	0.666667

dtype: float64

Figure 6.1. Accuracy results by user on the entire dataframe of red versus green questions using the Extra Trees Classifier.

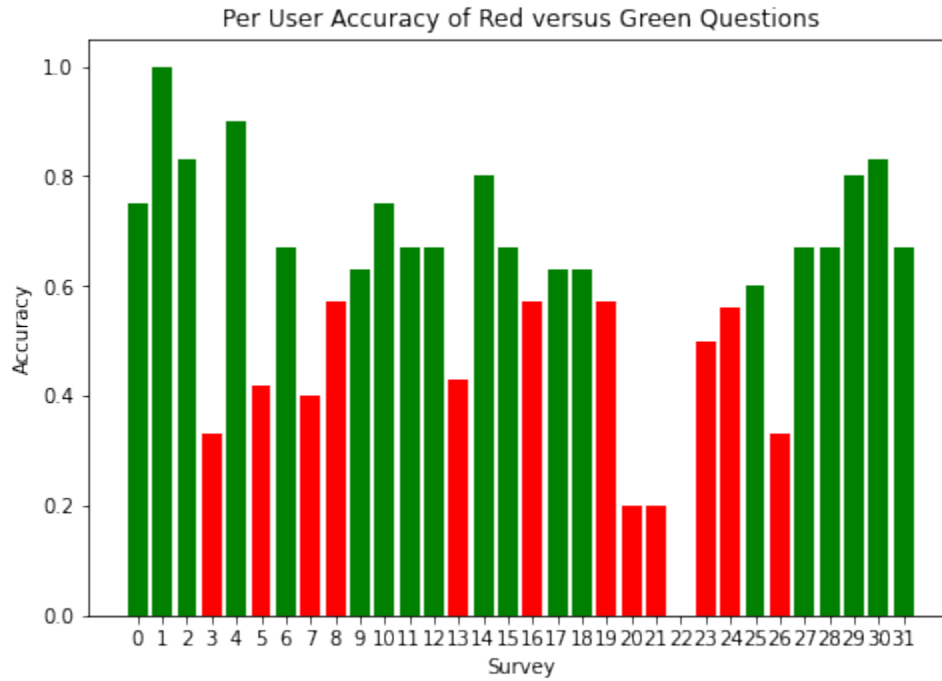


Figure 6.2. Accuracy results by user on the entire dataframe of red versus green questions using the Extra Trees Classifier.

Once we analyzed per-user accuracy of all question types, we further broke down the subset of questions into radio button questions only. Because the survey varied in question type, this prediction only left one green and one red question, thereby limiting the samples and the number of surveys in the test set. Out of the 15 predictions in the test set, the model correctly classified 13 surveys with 100.0% accuracy, one survey with 50.0% accuracy, and one survey incorrectly. This prediction was a binary classification of red versus green radio button questions and represents a scenario where a computer user answers the same type of expected question truthfully and deceptively, showing the potential to flag approximately 90% of deceptive users based on *velocity* and *angle* HCI inputs, specifically when filling out radio button submission questions on an online form. The per-user accuracy results on radio button questions only are depicted in Figure 6.3.

Based on Figure 6.3, survey 3 had a user accuracy of 0. When investigating the test predictions for survey 3, the question was labeled as “Red,” but predicted as “Green,” creating a FN predicament where the user should have been flagged for being deceptive,

but was rather classified as an honest user. This FN presents perhaps the most dangerous situation when relying on automated technology to determine threats on a network. A more detailed discussion of this topic can be found later in this chapter in Section 6.4.

```
In [776]: user_accuracy
Out[776]: survey_1      1.0
          survey_10     1.0
          survey_11     1.0
          survey_13     1.0
          survey_19     1.0
          survey_22     1.0
          survey_23     1.0
          survey_27     1.0
          survey_28     1.0
          survey_3      0.0
          survey_30     1.0
          survey_31     1.0
          survey_4      1.0
          survey_6      1.0
          survey_7      0.5
          dtype: float64
```

Figure 6.3. Accuracy results by user on the test set using Extra Trees Classifier and model set up discussed in Section 6.2.

Table 6.14 provides a comprehensive comparison of question 5 to question 21 for the subset of the test set based on Table 6.13 that was classified correctly. Questions 5 and 21 both ask the question, “What gender do you identify with?” However, question 5 is labeled as a green question, where the participant is asked to answer truthfully as themselves. In question 21, the participant is asked the exact same question, but asked to provide an answer while impersonating another person, thereby inducing some level of deception into their response.

The average *delta_time*, *velocity*, and *angle* are listed and, when comparing question 5 to question 21 per survey, the higher average (absolute value) of each feature per survey is highlighted in yellow. Out of the 13 surveys that were positively classified, 10 (76.92%) had a higher average *delta_time* value when answering truthfully (question 5), compared to

when acting in a deceptive manner (question 21). Average *velocity* was fairly even, as 6 out of 13 (46.15%) participants answered faster when telling the truth, and the remaining 53.85% of participants had higher average velocities when responding deceptively. However, when participants demonstrated greater velocities when telling the truth vice being deceptive, their responses were between 0.04 pixels/ms and 0.44 pixels/ms faster. On the other hand, when participants demonstrated quicker responses while being deceptive, their increase in average velocity covered a greater range, spanning between 0.01 pixels/ms and 0.62 pixels/ms. Survey 10 is an example of a participant who had a higher average velocity when deceiving than when telling the truth. Details of survey 10's velocity histograms, velocity profiles and standard deviations can be found in Section 6.3.1. On the other hand, 62% (8 out of 13) of participants had greater average angles when acting deceptively versus acting truthfully. Once again, survey 10 is an example case in which the participant recorded a higher average angle for question 21 than question 5.

Survey	Question	Avg. ΔTime	Avg. Velocity	Avg. Angle
Survey_1	5	46.33	1.45	-0.61
Survey_1	21	40.63	1.25	-0.67
Survey_4	5	50.74	0.87	-0.01
Survey_4	21	62.91	1.12	0.26
Survey_6	5	36.41	0.65	0.59
Survey_6	21	51.73	0.98	-0.28
Survey_11	5	44.70	1.04	-0.22
Survey_11	21	39.69	0.60	-0.36
Survey_13	5	30.98	0.67	-0.05
Survey_13	21	25.11	0.56	-0.57
Survey_19	5	11.64	0.98	-0.38
Survey_19	21	10.95	0.94	0.11
Survey_22	5	29.58	0.49	-0.50
Survey_22	21	18.00	0.37	0.00
Survey_23	5	23.96	1.33	0.11
Survey_23	21	21.19	1.54	-0.48
Survey_27	5	39.34	0.46	0.11
Survey_27	21	24.88	0.47	-0.35
Survey_28	5	34.46	0.64	-0.39
Survey_28	21	38.77	0.77	-0.42
Survey_29	5	23.10	0.51	-0.77
Survey_29	21	22.85	0.69	-0.89
Survey_30	5	30.65	0.95	-0.39
Survey_30	21	24.76	0.87	0.25
Survey_31	5	45.44	0.54	-0.69
Survey_31	21	31.09	1.16	-0.46

Table 6.14. Comprehensive average summary statistics for comparing green and red responses to radio button questions. The surveys included are ones that were correctly classified using the Extra Trees Classifier. Question 5 represents a green radio button question and question 21 is a red radio button question. The highest feature value for each survey is highlighted in yellow.

The standard deviations for each feature of the subset of surveys that were correctly classified can be found in Table 6.15. These standard deviations correspond to the average summary statistics found in Table 6.14. On average, both *delta_time* and *angle* produced greater standard deviations when responses were truthful than deceptive. From this subset, 76.92% of participants exhibited a higher standard deviation for *delta_time* when answering truthfully, compared to when acting deceptively. In contrast, *velocity* produced a greater standard deviation when participants were acting deceptively rather than truthful. Specifically, 7 out of 12 (58.33%) surveys yielded a higher standard deviation for *velocity* when being deceptive. Of note, survey 27 had the exact same standard deviation for both questions. An example of a survey that follows this pattern can be seen in survey 28's *velocity* histograms shown in Figure 6.15 in Section 6.3.1.

Out of the 13 surveys analyzed, 5 out of 13 (38.46%) participants had higher standard deviations of *angle* when answering deceptively. If the trajectory of mouse movement is a straight line, the *angle* measurement will remain consistent. However, if a user's mouse movement trajectory does not occur in a straight line, the measured *angle* will change. An example of how the changing mouse trajectory affects the *angle* measurement is demonstrated in Figure 6.4. Correlating with previous work done regarding mouse trajectory on insurance fraud, the greater standard deviation in *angle* may likely be attributed to the user's mouse paths following less direct routes to target destinations when deceiving than when being honest [3]. An example of how mouse trajectory affected angle specifically in our survey can be seen in survey 30's case study in Section 6.3.1, which details survey 30's *angle* histograms and *angle* profiles over time. The varied *angle* sizes and greater dispersion can be seen in the histograms in Figure 6.23, while the higher number of distinct peaks in part (b) of Figure 6.24 indicates more varied angles throughout mouse movement when being deceptive.

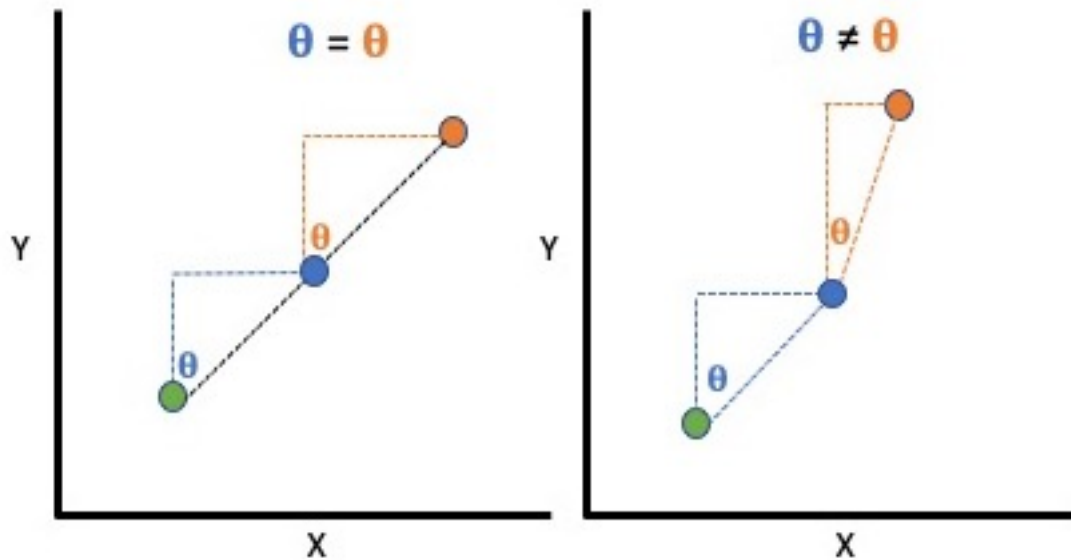


Figure 6.4. The left-hand graphic depicts a consistent mouse trajectory in which the cursor follows a straight line. In this case, the *angle* measurement does not change, as indicated by the blue and orange theta measurements. On the right-hand graphic, the mouse trajectory does not follow a straight line, thus creating a smaller *angle* measurement for the orange theta than the blue theta.

Survey	Question	S.D. ΔTime	S.D. Velocity	S. D. Angle
Survey_1	5	188.33	1.99	1.70
Survey_1	21	56.23	1.81	1.82
Survey_4	5	177.66	1.09	2.08
Survey_4	21	209.05	2.01	2.01
Survey_6	5	128.89	1.17	1.59
Survey_6	21	168.75	1.62	1.71
Survey_11	5	153.97	1.94	2.01
Survey_11	21	125.35	1.19	2.07
Survey_13	5	81.58	1.03	1.52
Survey_13	21	56.43	1.07	1.70
Survey_19	5	23.15	0.93	2.19
Survey_19	21	27.01	1.24	1.99
Survey_22	5	274.68	0.85	1.57
Survey_22	21	127.36	0.64	2.12
Survey_23	5	139.12	2.39	1.64
Survey_23	21	83.37	2.34	1.58
Survey_27	5	128.80	0.79	1.89
Survey_27	21	73.48	0.79	1.48
Survey_28	5	112.17	0.62	1.95
Survey_28	21	38.77	1.15	1.83
Survey_29	5	44.38	0.60	1.84
Survey_29	21	31.42	0.77	1.83
Survey_30	5	58.59	1.23	2.18
Survey_30	21	51.22	1.19	1.91
Survey_31	5	201.85	0.83	2.08
Survey_31	21	73.32	1.79	2.01

Table 6.15. Comprehensive standard deviation summary statistics for comparing green and red responses to radio button questions. The surveys included are ones that were correctly classified using the Extra Trees Classifier. Question 5 represents a green radio button question and question 21 is a red radio button question.

6.3.1 Individual Case Studies

Survey 10 Results

In order to view examples in which the classification was successful, the *velocity* and *angle* profiles were extracted from random surveys that were predicted accurately per Figure 6.3. Figure 6.5 reveals the *velocity* histograms for questions 5 and 21 (a green radio button question and red radio button question, respectively) from survey 10, while Figure 6.6 represents the *velocity* profiles for the same questions. For question 5, survey 10 had an average *velocity* of 0.461 pixels/ms. On the other hand, the average *velocity* for question 21 was 0.629 pixels/ms, 0.168 pixels/ms higher than the truth-telling response. While the average *velocity* for the impersonated answer was higher, the standard deviation was also greater. The standard deviation for *velocity* for question 5 was 0.593 pixels/ms and for question 21 it was 0.951 pixels/ms, accounting for a 35.8% increase in standard deviation.

Survey 10's Velocity Histograms

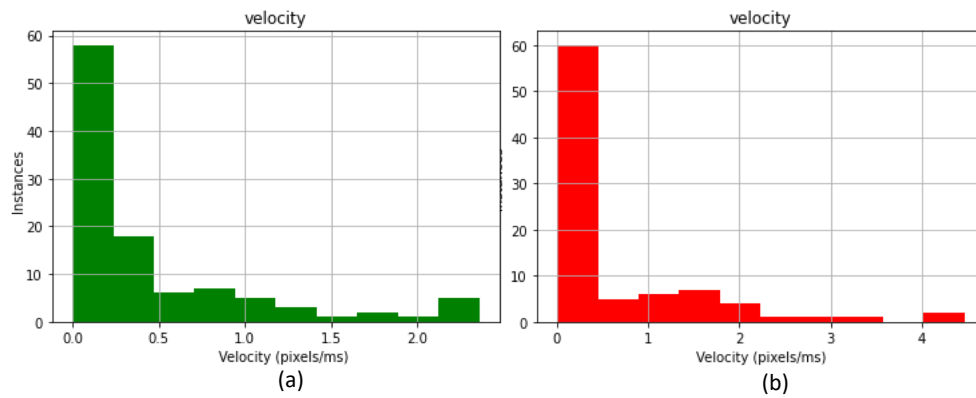


Figure 6.5. (a) Survey 10's velocity histogram for question 5, a green radio button question. (b) Survey 10's velocity histogram for question 21, a red radio button question.

Survey 10's Velocity Profiles

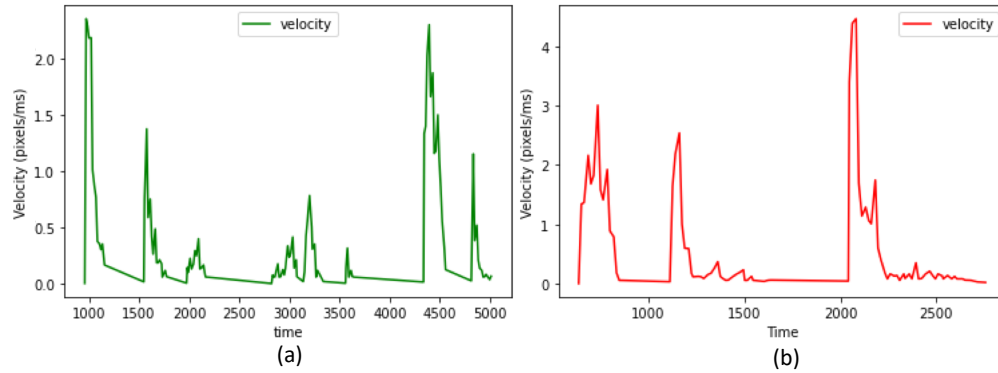


Figure 6.6. (a) Survey 10's velocity profile for question 5, a green radio button question. (b) Survey 10's velocity profile for question 21, a red radio button question.

Survey 10's *angle* histograms and profiles, seen in Figures 6.7 and 6.8, detail the change in the size of the *angle* produced by mouse movements when answering a single radio button style question, in both a truth-telling scenario and a deceptive scenario. The average *angle* for survey 10's question 5 was -0.42 radians and the average *angle* for question 21 was -0.60 radians, denoting more change in trajectory on average. However, the standard deviation for *angle* was 1.95 when answering truthfully, compared to a standard deviation of 1.51 when acting deceptively.

Survey 10's Angle Histograms

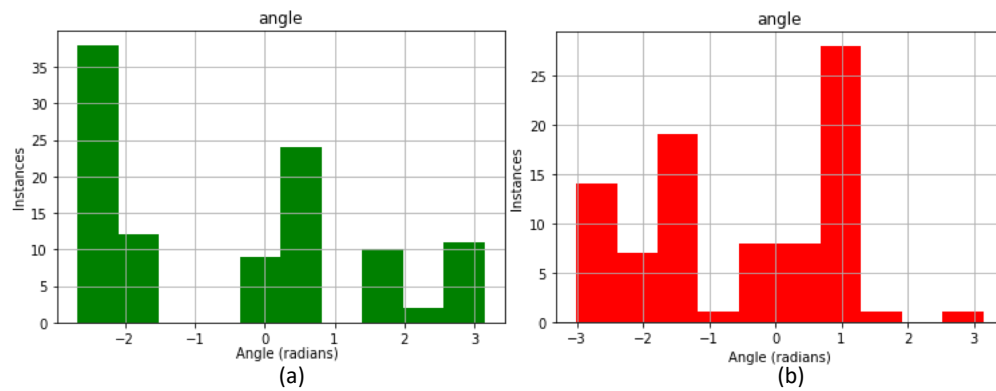


Figure 6.7. (a) Survey 10's angle histogram for question 5, a green radio button question. (b) Survey 10's angle histogram for question 21, a red radio button question.

Survey 10's Angle Profiles

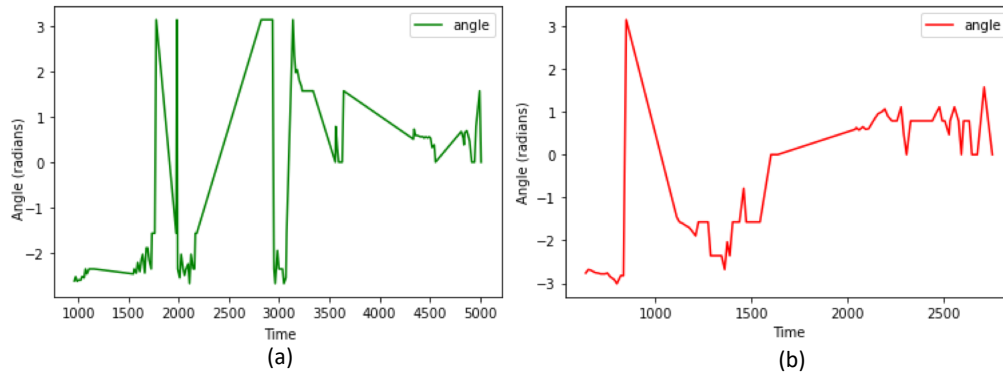


Figure 6.8. (a) Survey 10's angle profile for question 5, a green radio button question. (b) Survey 10's angle profile for question 21, a red radio button question.

Survey 13 Results

Survey 13's *velocity* histograms and *velocity* profiles are detailed in Figures 6.9 and 6.10. Survey 13 recorded an average *velocity* of 0.67 pixels/ms for question 5, compared to an average *velocity* of 0.56 pixels/ms for question 21. Standard deviation was greater for question 5 by approximately 4.6%. Looking at part (a) of Figure 6.10, the participant reached a much higher initial *velocity* at the start of the question as compared to part (b), where the participant reached about half the speed at the start of question 21. Since the user reached a higher initial *velocity* while telling the truth, this could potentially indicate a quicker response-activation model as detailed in Chapter 2. Additionally, the participant reached a much higher *velocity* at the tail-end of question 21 when compared to question 5, potentially indicating a desire to move on quickly from said deception.

Survey 13's Velocity Histograms

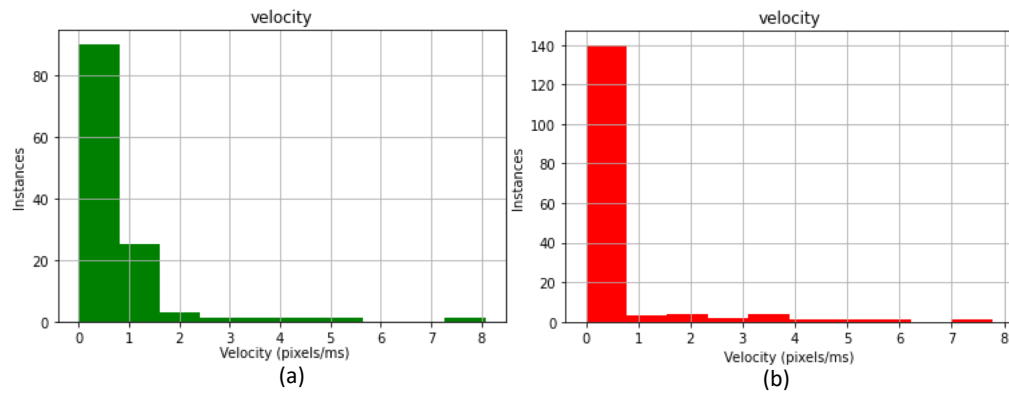


Figure 6.9. (a) Survey 13's velocity histogram for question 5, a green radio button question. (b) Survey 13's velocity histogram for question 21, a red radio button question.

Survey 13's Velocity Profiles

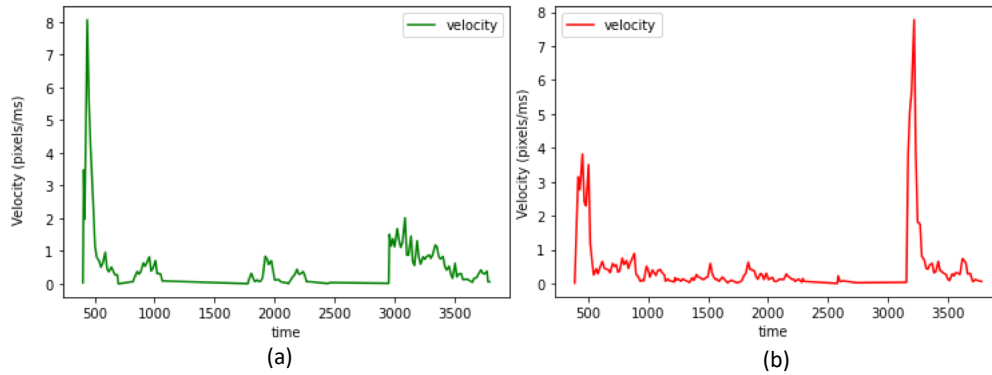


Figure 6.10. (a) Survey 13's velocity profile for question 5, a green radio button question. (b) Survey 13's velocity profile for question 21, a red radio button question.

The *angle* histograms and profiles for survey 13 are depicted in Figures 6.11 and 6.12. Similar to survey 10, survey 13 also recorded a higher (absolute value) average for question 21, compared to question 5. The average *angle* for question 5 was -0.06 radians, compared to question 21's average of -0.57 radians. Similarly, the standard deviation for *angle* was also approximately 20% greater during deception (question 21). *Angle* variation can be seen when comparing parts (a) and (b) of Figure 6.11, which shows more variation and dispersion of *angle* size in part (b).

Survey 13's Angle Histograms

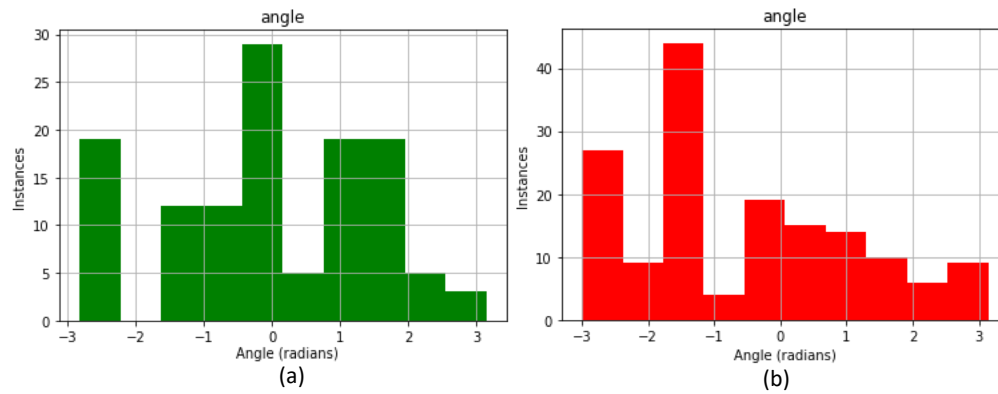


Figure 6.11. (a) Survey 13's angle histogram for question 5, a green radio button question. (b) Survey 13's angle histogram for question 21, a red radio button question.

Survey 13's Angle Profiles

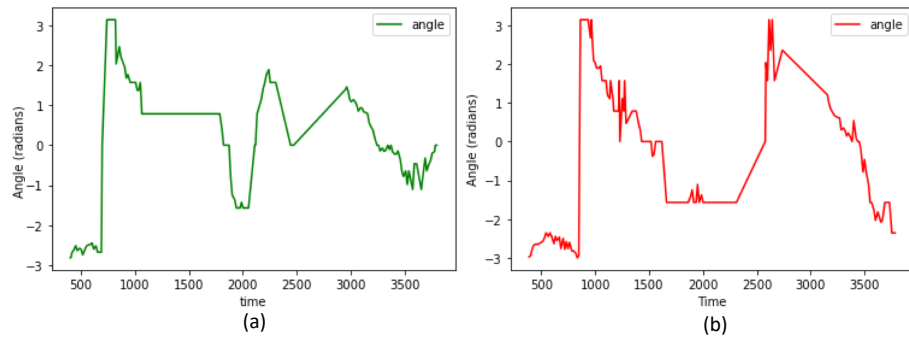


Figure 6.12. (a) Survey 13's angle profile for question 5, a green radio button question. (b) Survey 13's angle profile for question 21, a red radio button question.

The *delta_time* histograms for survey 13's question 5 and question 21 are depicted in Figure 6.13, while the *delta_time* profiles are depicted in Figure 6.14. The average change in time for question 5 was 30.98 ms, while the average change in time for question 21 was 25.11 ms. The standard deviation for question 5 was much higher at 81.58 ms, compared to question 21's standard deviation of 56.43 ms, potentially indicating more drastic and distinct mouse movements when telling the truth.

Survey 13's Change in Time Histograms

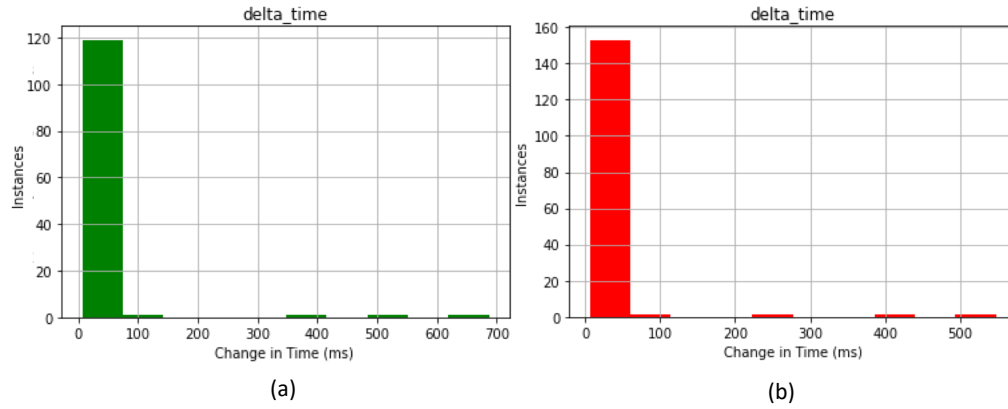


Figure 6.13. (a) Survey 13's change in time histogram for question 5, a green radio button question. (b) Survey 13's change in time histogram for question 21, a red radio button question.

Survey 13's Change in Time Profiles

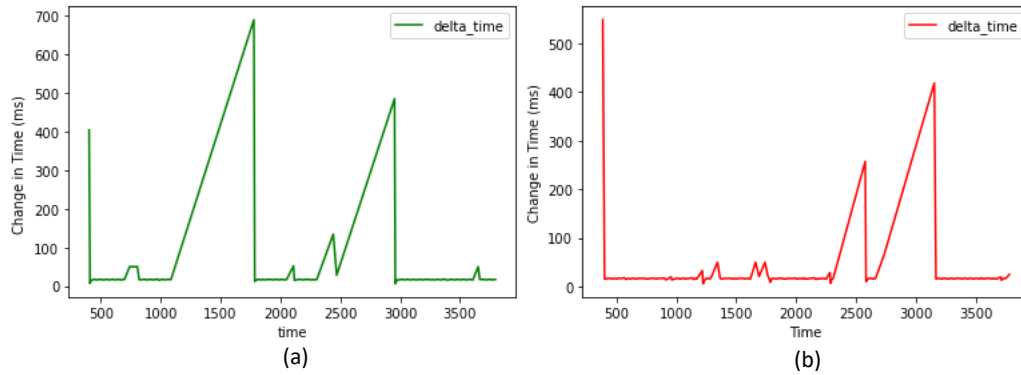


Figure 6.14. (a) Survey 13's change in time profile for question 5, a green radio button question. (b) Survey 13's change in time profile for question 21, a red radio button question.

Survey 28 Results

Survey 28's *velocity* histograms are depicted in Figure 6.15. Survey 28 had an average *velocity* of 0.64 pixels/ms while answering question 5, compared to an average *velocity* of 0.77 pixels/ms when answering question 21. As seen in the histograms, question 21 had a standard deviation of 1.15 pixels/ms, while question 5's standard deviation was approximately half of that, at 0.62 pixels/ms. Survey 28's *velocity* profiles are depicted in Figure 6.16. As indicated by the value on the y-axis, the participant increased *velocity* in a similar manner at the head and tail end of the response for both questions, but by approximately 3 orders of magnitude greater when deceiving, depicted in part (b).

Survey 28's Velocity Histograms

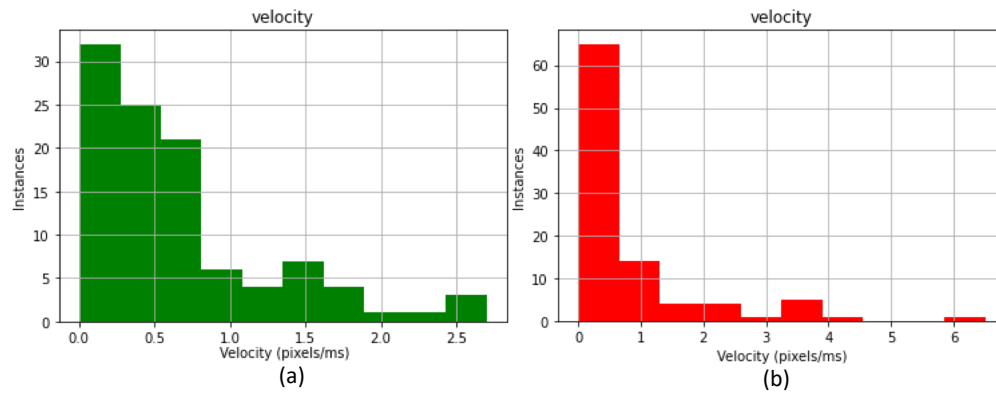


Figure 6.15. (a) Survey 28's velocity histogram for question 5, a green radio button question. (b) Survey 28's velocity histogram for question 21, a red radio button question.

Survey 28's Velocity Profiles

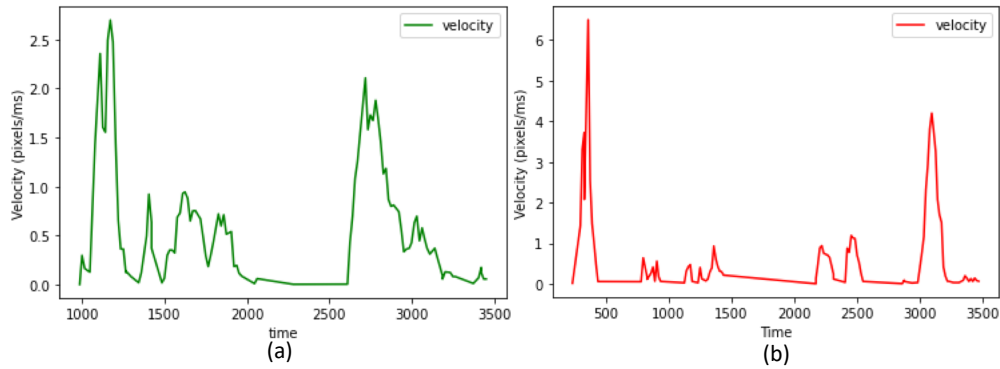


Figure 6.16. (a) Survey 28's velocity profile for question 5, a green radio button question. (b) Survey 28's velocity profile for question 21, a red radio button question.

Survey 28's *angle* histograms and *angle* profiles are illustrated in Figure 6.17 and Figure 6.18, respectively. Survey 28 had a greater average *angle* when acting deceptively, recording a -0.42 radian *angle*, compared to an average -0.39 radian *angle* when telling the truth in question 5. Additionally, survey 28 utilized approximately three times more 0 radian angles when telling the truth, vice when deceiving. When comparing parts (a) and (b) of Figure 6.18, a greater amount of time is spent using larger angles on response 21 (b) than in part (a). *Angle* profiles for questions 5 and 21 follow a similar format, but the beginning portion of part (b) depicts many more drastic changes in angle than part (a).

Survey 28's Angle Histograms

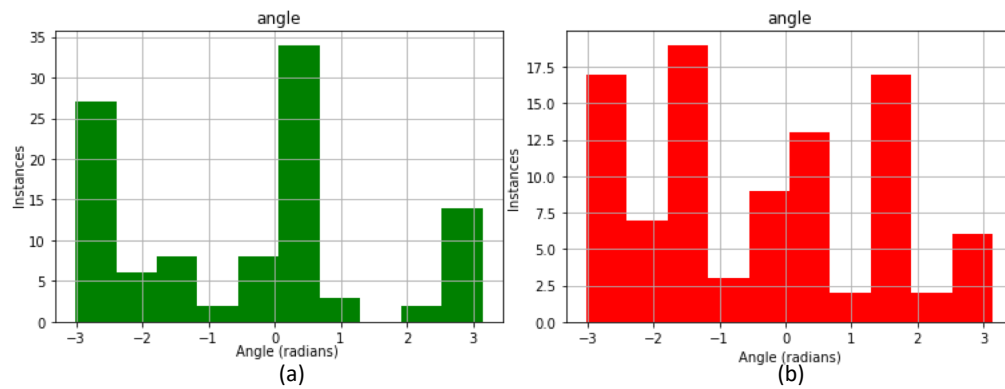


Figure 6.17. (a) Survey 28's angle histogram for question 5, a green radio button question. (b) Survey 28's angle histogram for question 21, a red radio button question.

Survey 28's Angle Profiles

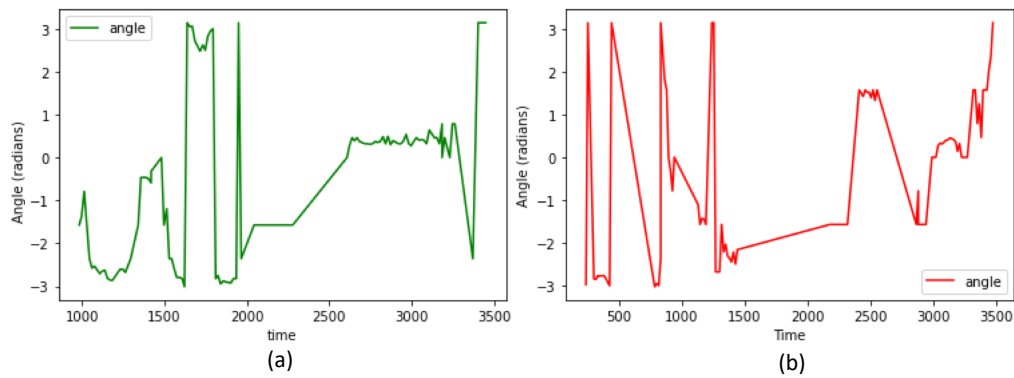


Figure 6.18. (a) Survey 28's angle profile for question 5, a green radio button question. (b) Survey 28's angle profile for question 21, a red radio button question.

For change in time, survey 28's associated histograms and *delta_time* profiles are detailed in Figures 6.19 and 6.20. Survey 28 recorded an average *delta_time* of 34.46 ms for question 5, compared to an average *delta_time* of 38.77 ms for question 21. However, standard deviation was actually higher for question 5 at 112.17 ms, compared to a standard deviation of 95.56 ms for question 21.

Survey 28's Change in Time Histograms

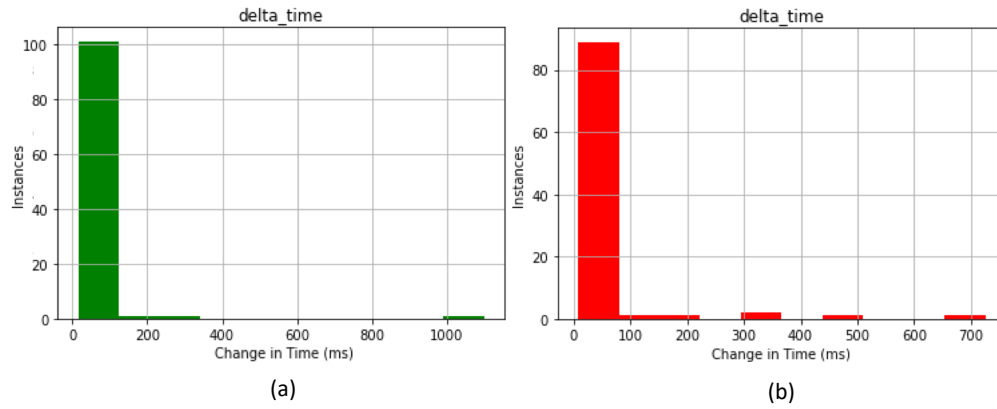


Figure 6.19. (a) Survey 28's change in time histogram for question 5, a green radio button question. (b) Survey 28's change in time histogram for question 21, a red radio button question.

Survey 28's Change in Time Profiles

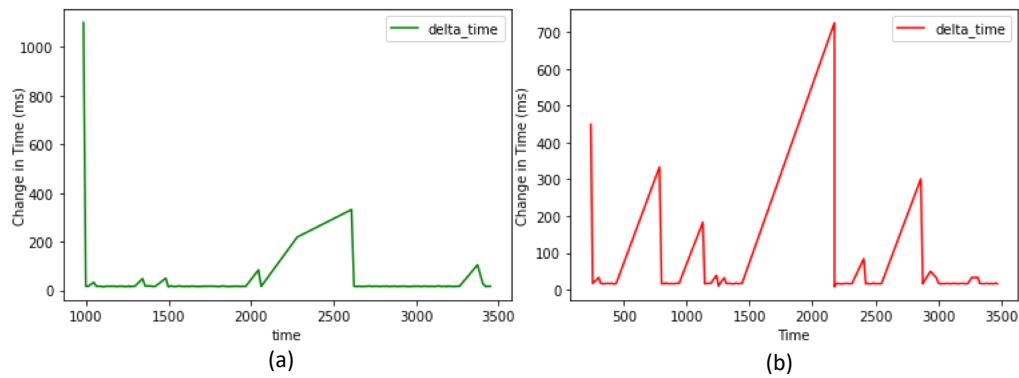


Figure 6.20. (a) Survey 28's change in time profile for question 5, a green radio button question. (b) Survey 28's change in time profile for question 21, a red radio button question.

Survey 30 Results

The *velocity* histograms and profiles for survey 30, another survey that was classified correctly, are seen in Figures 6.21 and 6.22. Similar to survey 13, survey 30 also showed a higher average *velocity* when telling the truth than when acting deceptively, similar to survey 28. The average *velocity* for question 5 was 0.949 pixels/ms, while the average *velocity* for question 21 was 0.872 pixels/ms. Also unlike survey 10's results, survey 30 actually had a lower standard deviation for *velocity* for question 21, but only by approximately 4%. Four distinct *velocity* peaks are noticeable in question 21's *velocity* profile, whereas question 5 has one distinct *velocity* peak.

Survey 30's Velocity Histograms

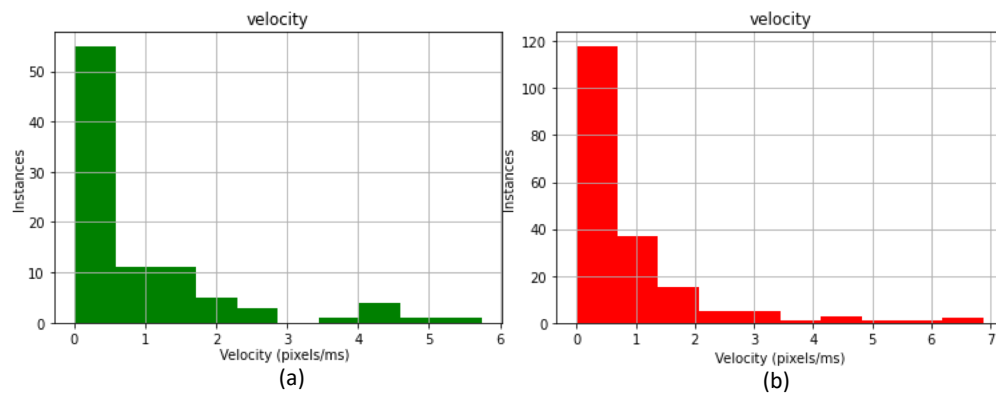


Figure 6.21. (a) Survey 30's velocity histogram for question 5, a green radio button question. (b) Survey 30's velocity histogram for question 21, a red radio button question.

Survey 30's Velocity Profiles

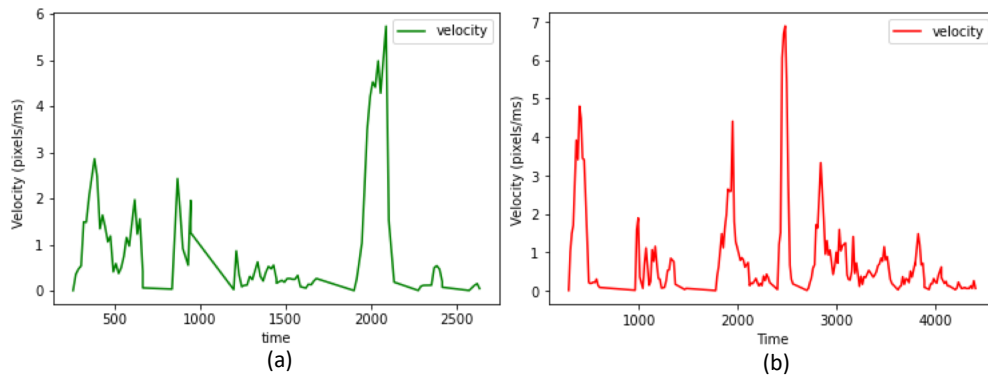


Figure 6.22. (a) Survey 30's velocity profile for question 5, a green radio button question. (b) Survey 30's velocity profile for question 21, a red radio button question.

For *angle*, survey 30's question 5 had an average of -0.39 radians, while question 21 yielded an average *angle* of 0.25 radians. Of note, this is the first case study in which the average *angle* for question 21 has been smaller than that of question 5. Question 5 also exhibited a higher standard deviation for *angle*. Of note, survey 30's *angle* histogram in part (b) of Figure 6.23 depicts approximately 80 instances of -3 or 3 radian angles, whereas part (a) depicts approximately 48 instances of similar angles. This may indicate that the user used less drastic *angle* changes when answering truthfully than when answering deceptively. Additionally, the angle profile in Figure 6.24 depicts more peaks in part (b) than in part (a).

Survey 30's Angle Histograms

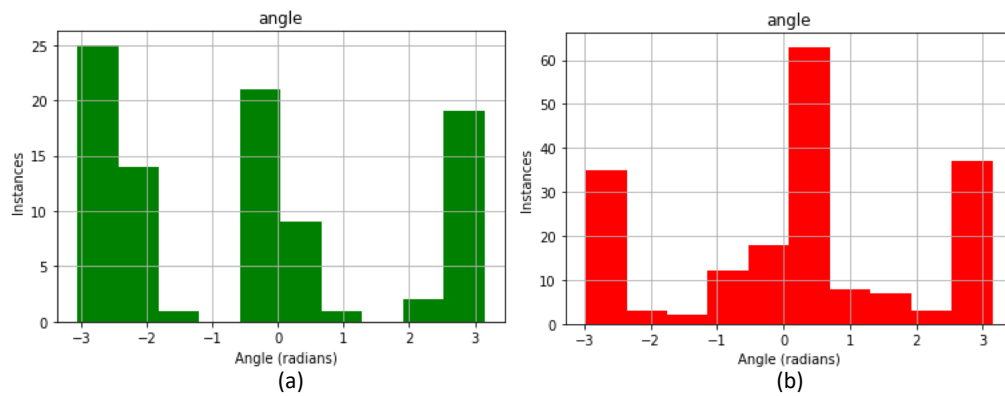


Figure 6.23. (a) Survey 30's angle histogram for question 5, a green radio button question. (b) Survey 30's angle histogram for question 21, a red radio button question.

Survey 30's Angle Profiles

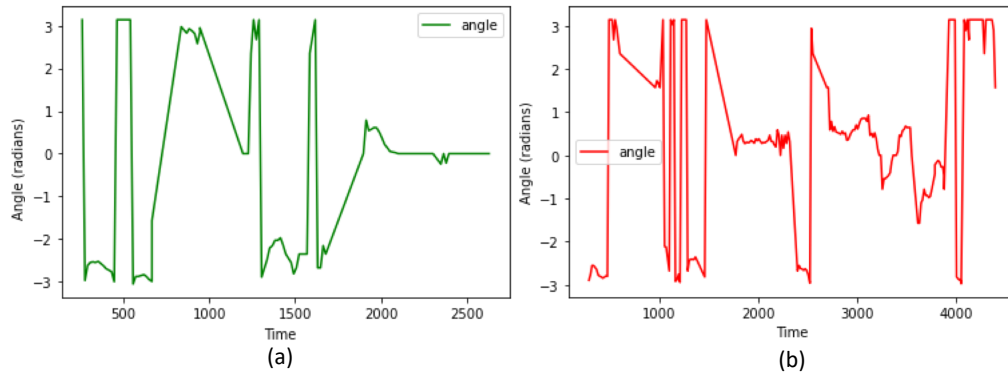


Figure 6.24. (a) Survey 30's angle profile for question 5, a green radio button question. (b) Survey 30's angle profile for question 21, a red radio button question.

6.4 Discussion

6.4.1 Aspects of HCI for Deception Detection

The best model results were produced by using *velocity* and *delta_time* in conjunction. In combination, using *velocity* and *delta_time* as features in an Extra Trees classifier increased baseline accuracy by a total of 37.41%, by far the best results out of the various combinations analyzed. *Velocity* and *angle* together recorded the next highest increase in accuracy, boosting model accuracy by about 26.89%. As a sole feature, *delta_time* and *angle* provided the highest increases in accuracy, each with 21.62%, followed by *velocity* with a 16.36% increase in accuracy.

While *angle* and *delta_time* performed the best when classifying green versus red questions,

all three features in conjunction produced the best results when classifying green versus orange questions. Although *delta_time* produced the highest accuracy as a sole feature when classifying green versus red questions, it was the only sole feature to decrease accuracy for green versus orange questions. The feature with the single greatest influence on green versus orange classification was *velocity*, increasing accuracy by about 22.77

This research shows that *angle*, along with *velocity*, and *delta_time* can be used as HCI indicators for deception detection. Based on how *angle* was calculated in this study, the *angle* measurement most likely corresponds with mouse trajectory studies, discussed in Chapter 2. Shown in Figure 6.4, a straighter mouse trajectory indicates smaller changes in angle, as represented by horizontal lines in the *angle* profiles. Smaller changes in angle, results in a smaller standard deviation for *angle*. Similar to studies that analyzed deception's effect on mouse trajectory, the response activation model accounts for larger *angle* deviations when lying, as a user's mouse path towards a destination may not take the shortest route when acting in a deceptive manner [6] [29]. Examples of how mouse movement and trajectory correlate to deception detection can be found in Figures 2.2 and 3.1.

While a higher percentage of users showed greater average *velocity* when answering deceptively than when answering truthfully, it is important to note the location of questions. According to the survey layout explained in Chapter 3, every survey followed an identical format. For the first half of questions, participants were asked to answer questions as themselves. For the second half of questions, participants were asked to impersonate a close friend or significant other. Since the impersonation questions always followed as the second portion of the survey, there is some level of familiarity with the survey format that could be indicative of increases in average *velocity* for survey participants while being deceptive. In order to create more accurate results that limit the effects of familiarity on response speed, surveys should be randomized and given in varied orders.

In order to create a more balanced dataset with a larger sample size, more radio button questions should be added. Focusing on one type of question response would allow potentially better modeling than relying on varied question types.

6.4.2 False Positives and False Negatives

When comparing green and red questions in isolation, FNs correlate to a red question being falsely predicted as a green question. This FN indicates a deceptive response that was then automatically detected as an honest response. While FPs can also occur (predicting a labeled green question as a red question), this FN scenario presents a deeper security concern. A FN situation induces error into a potential system of automated detection of deceptive behavior, whereby a deceptive user may gain entry into a network or system without proper detection. A FP scenario is also possible, which would indicate an honest user being flagged and filtered as a deceptive user. If an automated system detects and flags an honest user as a deceptive user, a legitimate person may lose access to a necessary network, incur a heavier workload to unlock accounts, and may feel embarrassment or anger towards the organization. While FPs certainly are not ideal, creating a model that errs towards FPs rather than FNs would be ideal, as filtering out threatening personnel is a higher priority than ensuring quick access to all legitimate personnel. To thoroughly protect networks and accounts, any automated detection system should be reinforced with other methods, such as human corroboration. In many instances throughout the DOD, this is already the case. A DOD worker who attempts to log in or use a government computer to gain access to a specified network has most likely already undergone screening or background checks. If proper identification checks are in place prior to computer use, HCI techniques provide additional methods of verification to ensure malicious activity on networks is limited.

6.4.3 Malicious Actors

While utilizing HCI for deception detection provides many benefits, it can also potentially be altered to avoid detection altogether. Using HCI elements for deception detection is non-intrusive, can be collected on user's consistently, and can be used to create portions of an intrusion detection system. However, if knowledge of this methodology is known to malicious actors, such threats might purposely alter or train their mouse movements to mimic consistent truth-like responses. In the same way that threat actors attempt to evade barriers already in place for detection such as polygraphs, they will indeed attempt to learn and alter behavior in order to evade detection through HCI. Countermeasures and awareness of this distinct possibility are imperative as researchers create detection systems based on HCI functionalities.

CHAPTER 7:

Conclusions and Future Work

This chapter provides direction for potential future and tangential research on HCI-based insider threat detection and concludes with a summary of the accomplishments and contributions of this study.

7.1 Future Work

Our study eventually focused on the binary classification of basic green versus red radio button question responses. However, because the focus of the study did not become clear until after data collection, the sample size was limited. Future work should collect many green versus red radio button responses in order to allow deeper analysis of the three features discussed in this study.

Future work should also isolate radio button questions of colors yellow and orange as well, where “unexpected” questions are asked to a user who is both supposed to answer some of these questions truthfully and others as an impersonated person. Our study briefly modeled unexpected questions that were asked to the impersonating person (orange), but did not explore unexpected questions that were asked of an honest user (yellow) in detail, due to limited sample size. This study demonstrated the usefulness of unexpected questions, as binary classifications that included unexpected question comparisons largely outperformed those that did not.

Rather than simple binary classifications, future work should consider multi-classification as well. The ability to distinguish not only green responses (expected biographical truths) from red responses (expected biographical lies), but green and red from yellow and orange questions (unexpected questions) provides insights on the types of questions to include in online forms that could potentially indicate deception. A multi-classification would be beneficial in terms of categorizing the various types of deception, such as ill-intentioned deception, benign user deception, or even embarrassment-driven deception to protect personal information and integrity.

Lastly, future work should be done to analyze other HCI features on text-based responses that could mitigate the need for a user-specific model in deception detection based on keyboard inputs. In addition to analyzing text-based responses, other multiple choice responses, such as selection or checkboxes, should also be analyzed. Additional HCI features should be viewed as potential indicators of deception.

7.2 Contributions and Conclusion

Our study verified the potential for using HCI cues as a method of deception detection within networks. Aspects of related work discussed in Chapter 2 are also corroborated within our study. Although possible rationales are discussed in Section 6.4, 10% more participants had higher average velocities when acting deceptively than when acting in a truthful manner. In a study involving gyroscopes and accelerometers to determine how people orient their bodies in response and in preparation for deception, participants provided evidence that liars display higher rates of mouse acceleration and rotation in their responses than their truth-telling counterparts [11]. While *velocity* is a different measurement than the rotation rate explored in [11] and our study focused on HCI rather than general hand movement, higher rates of average velocity were seen during deceptive responses. Moreover, the higher average *angle* observations are consistent with higher rates of rotation seen in deceptive users during the gyroscope and accelerometer study [11]. In the future, attention should be paid to *angle* and how deception manifests through it.

Though not a main focus in our study, the effectiveness of introducing “unexpected” questions to determine deception is also validated. In our study, yellow and orange questions both represented “unexpected” questions. Yellow questions were applied to the subset of questions that were asked to truthful participants, while orange questions applied to the subset of questions that were asked to participants who were impersonating another user. In every basic model before major parameter hyper-tuning (including models that included all question types), the binary classification of green versus yellow questions performed better than the binary classification of green versus red questions. These results indicate a noticeable difference to responses to expected and unexpected questions. Additionally, red versus orange binary classifications also saw increases in accuracy when compared to baseline accuracy, showcasing the noticeable difference in responses when introducing unexpected questions to an impersonated scenario. Lastly, utilizing all three features (*velocity*,

angle, and *delta_time*), classification of green versus orange questions showed a 13.29% increase in accuracy from its associated baseline accuracy. Green versus orange classification implies a scenario that compares an honest user encountering an expected question to an impersonated user encountering an unexpected question.

Our research added to the works discussed above and to the studies discussed in Chapter 2 primarily by verifying *angle* as a suitable HCI feature for deception detection and by proving the potential for a non-user specific model for deception classification. Utilizing the overall results in Chapter 6 and specific discussion points in Section 6.3, we showed that, out of the three features analyzed, *angle* was the most consistent feature in determining deception.

While previous studies focused on user-specific models to determine *when* and *if* an individual user was applying deception on a device, our study shifted focus in order to determine if a model could accurately detect deception in a user based on overall HCI features, rather than relying on a specific comparison of that user's current keyboard and mouse interaction to a historical log of that user's interaction. In short, our study proved that a non user-specific model could be applied to determine deception to some degree in a first-time user. Negating the need for a user-specific model would enable a more complete first-line of defense against malicious activity among networks and in the use of online forms where historical user data is unavailable. Preventing malicious activity at the earliest point of entry or on the periphery prior to access being granted is a high priority for businesses, schools, the government, and the DOD. However, malicious insiders are a well known threat, so, although preventing malicious activity on the periphery is a goal, it must be complimented by other defensive techniques. Utilizing non-user specific models may also play a relevant part in detecting insider threats on a network.

Implementing non-user specific models that achieve any level of accuracy greater than baseline accuracy would improve automated deception detection methods and thereby limit the threat axis on specific networks, websites, and computers by detecting malicious or untruthful behavior based on HCI components. Though there is much research to be done in this area, this work solidifies the possibility of utilizing a user's own HCI metrics in order to determine the credibility of their responses. Applications of this automated detection methodology span many workplaces and uses, including detecting impersonation of online test-taking atmospheres, detecting fraudulent behavior on high-risk websites such

as banking, homeowners, or education websites, as well as potential for deception detection on account creation in order to limit insider threat access. Including more automation into deception detection and thereby increasing recognition of certain deceptive and malicious activities can trigger mitigation efforts, therefore limiting the overall malicious activity on a network. Additionally, automating deception detection will reduce the overall manual effort needed for internal security. Once deception is detected, various methods could be put in place to immediately halt activity, such as screen locks, account deactivation, or an alarm being set off. Utilizing HCI as a form of deception detection allows the DOD to harness all available tools in order to limit security discrepancies and create a more robust and secure network.

APPENDIX A:

Dictionary Mapping of Web-Logged Events

Table A.1. Dictionary mapping of computer keyboard inputs to Numeric Codes.

Number Code	Input
0	Unknown
8	Backspace
9	Tab
13	Enter
16	Shift
17	Ctrl
18	Alt
19	Pause
20	Caps Lock
27	Escape
32	Space
33	Page Up
34	Page Down
35	End
36	Home
37	Left
38	Up
39	Right
40	Down
44	Print Screen
45	Insert
46	Delete
48	0

Continued on next page

Table A.1 – continued from previous page

Number Code	Input
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9
59	Semicolon
61	Equals
65	a
66	b
67	c
68	d
69	e
70	f
71	g
72	h
73	i
74	j
75	k
76	l
77	m
78	n
79	o
80	p
81	q
82	r
83	s

Continued on next page

Table A.1 – continued from previous page

Number Code	Input
84	t
85	u
86	v
87	w
88	x
89	y
90	z
91	Windows Start
93	Windows Menu
96	Numpad 0
97	Numpad 1
98	Numpad 2
99	Numpad 3
100	Numpad 4
101	Numpad 5
102	Numpad 6
103	Numpad 7
104	Numpad 8
105	Numpad 9
106	Numpad Multiply
107	Numpad Add
109	Numpad Subtract
110	Numpad Decimal
111	Numpad Divide
112	f1
113	f2
114	f3
115	f4
116	f5
117	f6

Continued on next page

Table A.1 – continued from previous page

Number Code	Input
118	f7
119	f8
120	f9
121	f10
122	f11
123	f12
144	Num Lock
145	Scroll Lock
173	Mute
174	Volume Down
175	Volume Up
186	Semicolon
187	Equals
188	Comma
189	Dash
190	Period
191	Slash
192	Back Quote
219	Open Bracket
220	Back Slash
221	Close Bracket
222	Quote

APPENDIX B:

Complete List of Survey Questions

1. Baseline Text: "Please type the following in the text box below: I promise to answer the questions truthfully and to the best of the ability, without help from the internet. I understand the effects of lying on a questionnaire for research and the negative consequences that can have undermining a study."
2. What is your first name?
3. What is your middle name?
4. What elementary school did you attend?
5. What gender do you identify with?
6. What is the current year? (Example: 1997)
7. In what city/town were you born?
8. What was the make of the first car you owned?
9. What month were you born?
10. What is your horoscope sign?
11. What is your birth date? (Mon/Day/Year, example: 11/2/1997)
12. How old are you? (Example: 21)
13. How many alcoholic drinks do you have weekly?
14. Have you ever smoked marijuana?
15. What is the name of your first love?
16. Who do you trust more than anyone in this world?
17. "For the remaining set of questions, please pick a significant other, parent, or close friend to impersonate. You should choose the person whom you know the most information about and therefore are most likely to be able to answer the questions truthfully and accurately. Please specify from which category below your new identity belongs and answer the questions to the best of your ability. Good luck!"
18. What is your first name?
19. What is your middle name?
20. What elementary school did you attend?
21. What gender do you identify with?

22. What country do you live in? (No abbreviations)
23. In what city/town were you born?
24. What was the make of the first car you owned?
25. What month were you born?
26. What is your horoscope sign?
27. What is your birth date? (Mon/Day/Year, example: 11/2/1997)
28. How old are you? (Example: 21)
29. How many alcoholic drinks do you have weekly?
30. Have you ever smoked marijuana?
31. What is the name of your first love?
32. Who do you trust more than anyone in this world?

List of References

- [1] J. K. Burgoon and D. B. Buller, “Interpersonal deception theory,” *Communication Theory*, vol. 6, no. 3, pp. 311–328, 1996. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1468-2885.1996.tb00132.x>
- [2] R. E. Carrion, J. P. Keenan, and N. Sebanz, “A truth that’s told with bad intent: An erp study of deception,” *Cognition*, vol. 114, no. 1, pp. 105–110, 2010.
- [3] J. L. Jenkins, J. Proudfoot, J. S. Valacich, G. M. Grimes, and J. F. N. Jr., “Sleight of hand: Identifying concealed information by monitoring mouse-cursor movements,” *J. Assoc. Inf. Syst.*, vol. 20, no. 1, p. 3, 2019. Available: <https://aisel.aisnet.org/jais/vol20/iss1/3>
- [4] W. D. Heaven. MIT Technology Review. [Online]. Available: <https://www.technologyreview.com/2020/05/11/1001563/covid-pandemic-broken-ai-machine-learning-amazon-retail-fraud-humans-in-the-loop>. Accessed Oct. 21, 2020.
- [5] Y. Pratheepan, J. V. Condell, and G. Prasad, “Individual identification from video based on" behavioural biometrics",” in *Behavioral biometrics for human identification: Intelligent applications*. IGI Global, 2010, pp. 75–100.
- [6] P. Zimmermann, S. Guttormsen, B. Danuser, and P. Gomez, “Affective computing—a rationale for measuring mood with mouse and keyboard,” *International journal of occupational safety and ergonomics*, vol. 9, no. 4, pp. 539–551, 2003.
- [7] C. Epp, M. Lippold, and R. L. Mandryk, “Identifying emotional states using keystroke dynamics,” in *Proceedings of the International Conference on Human Factors in Computing Systems, CHI 2011, Vancouver, BC, Canada, May 7-12, 2011*, D. S. Tan, S. Amershi, B. Begole, W. A. Kellogg, and M. Tungare, Eds. ACM, 2011, pp. 715–724. Available: <https://doi.org/10.1145/1978942.1979046>
- [8] K. Hill. New York Times. [Online]. Available: <https://www.nytimes.com/2020/01/18/technology/clearview-privacy-facial-recognition.html>. Accessed Jan. 6, 2021.
- [9] C. McKinsty, R. Dale, and M. J. Spivey, “Action dynamics reveal parallel competition in decision making,” *Psychological Science*, vol. 19, no. 1, pp. 22–24, 2008.
- [10] J. Freeman, R. Dale, and T. Farmer, “Hand in motion reveals mind in motion,” *Frontiers in psychology*, vol. 2, p. 59, 2011.

- [11] J. S. Valacich, J. L. Jenkins, M. D. Byrd, D. Kim, M. Kumar, and P. A. Williams, "Using accelerometer and gyroscope data in common mobile devices to assess credibility," in *26th Americas Conference on Information Systems, AMCIS 2020, Virtual Conference, August 15-17, 2020*, B. B. Anderson, J. Thatcher, R. D. Meservy, K. Chudoba, K. J. Fadel, and S. Brown, Eds. Association for Information Systems, 2020. Available: https://aisel.aisnet.org/amcis2020/adv_info_systems_research/adv_info_systems_research/25
- [12] M. Monaro, R. Spolaor, Q. Li, M. Conti, L. Gamberini, and G. Sartori, "Type me the truth!: Detecting deceitful users via keystroke dynamics," in *Proceedings of the 12th International Conference on Availability, Reliability and Security, Reggio Calabria, Italy, August 29 - September 01, 2017*. ACM, 2017, pp. 60:1–60:6. Available: <https://doi.org/10.1145/3098954.3104047>
- [13] M. Hibbeln, J. L. Jenkins, C. Schneider, J. S. Valacich, and M. Weinmann, "Investigating the effect of insurance fraud on mouse usage in human-computer interactions," in *Proceedings of the International Conference on Information Systems - Building a Better World through Information Systems, ICIS 2014, Auckland, New Zealand, December 14-17, 2014*, M. D. Myers and D. W. Straub, Eds. Association for Information Systems, 2014. Available: <http://aisel.aisnet.org/icis2014/proceedings/HCI/8>
- [14] M. M. Eloff and J. H. P. Eloff, "Human computer interaction: An information security perspectives," in *Security in the Information Society: Visions and Perspectives, IFIP TC11 17th International Conference on Information Security (SEC2002), May 7-9, 2002, Cairo, Egypt* (IFIP Conference Proceedings), A. Ghonaimy, M. T. El-Hadidi, and H. K. Aslan, Eds. Kluwer, 2002, vol. 214, pp. 535–546.
- [15] K. Hornbæk and A. Oulasvirta, "What is interaction?" in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Denver, CO, USA, May 06-11, 2017*, G. Mark, S. R. Fussell, C. Lampe, m. c. schraefel, J. P. Hourcade, C. Appert, and D. Wigdor, Eds. ACM, 2017, pp. 5040–5052. Available: <https://doi.org/10.1145/3025453.3025765>
- [16] A. F. M. N. H. Nahin, J. M. Alam, H. Mahmud, and M. K. Hasan, "Identifying emotion by keystroke dynamics and text pattern analysis," *Behav. Inf. Technol.*, vol. 33, no. 9, pp. 987–996, 2014. Available: <https://doi.org/10.1080/0144929X.2014.907343>
- [17] A. Rizzo and M. Palmonari, "Context and consciousness: Activity theory and human computer interaction, bonnie a. nardi (ed.)," *User Model. User Adapt. Interact.*, vol. 8, no. 1-2, pp. 153–157, 1998. Available: <https://doi.org/10.1023/A:1008207414131>

- [18] A. Niesler and G. Wydmuch, “User profiling in intelligent tutoring systems based on myers-briggs personality types,” in *Proceedings of the international multicongference of engineers and computer scientists*, 2009, vol. 1.
- [19] M. T. Robinson. The Career Planner. [Online]. Available: <https://https://www.careerplanner.com/MB2/TypeInPopulation.cfm>. Accessed Nov. 15, 2020.
- [20] 16Personalities. [Online]. Available: <https://www.16personalities.com/isfj-personality>. Accessed Nov. 15, 2020.
- [21] J. Zou and L. Schiebinger. Nature. [Online]. Available: <https://www.nature.com/articles/d41586-018-05707-8>. Accessed Apr. 21, 2020.
- [22] M. Monaro, I. Zampieri, G. Sartori, P. Pietrini, and G. Orrù, “The detection of faked identity using unexpected questions and choice reaction times,” *Psychological Research*, pp. 1–9, 2020.
- [23] M. D. Byrd, J. L. Jenkins, J. S. Valacich, and P. A. Williams, “Creating a realistic experimental scenario for hci-based deception detection research with ground truth and unsanctioned malicious acts,” 2018.
- [24] M. W. Docs, “Date.now().” Available: [://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/now](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date/now)
- [25] T. A. Salthouse, “Effects of practice on a typing-like keying task,” *Acta psychologica*, vol. 62, no. 2, pp. 189–198, 1986.
- [26] W. Koehrsen. Towardsdatascience. [Online]. Available: <https://towardsdatascience.com/histograms-and-density-plots-in-python-f6bda88f5ac0>. Accessed May 3, 2021.
- [27] J. Kho. Towardsdatascience. [Online]. Available: <https://towardsdatascience.com/why-random-forest-is-my-favorite-machine-learning-model-b97651fa3706>. Accessed May 3, 2021.
- [28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [29] T. N. Welsh and D. Elliott, “Movement trajectories in the presence of a distracting stimulus: Evidence for a response activation model of selective reaching,” *The Quarterly Journal of Experimental Psychology Section A*, vol. 57, no. 6, pp. 1031–1057, 2004.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California