Received: 22 February 2025 | Revised: 8 April 2025 | Accepted: 23 April 2025 | Published online: 8 May 2025

RESEARCH ARTICLE

Smart Wearable Technology
2025, Vol. 1, A10, 7 Pages
DOI: 10.47852/bonviewSWT52025469

BON VIEW PUBLISHING

Heuristic Sentiment Analysis for Social Engineering Mitigation During Interactive Immersion with Smart Wearable Technology

Sarah Katz[1,*]

[1]*Department of Cyberpsychology, Capitol Technology University, USA*

Abstract: Distraction caused by the visual processing of multiple objects during augmented reality (AR) and other forms of interactive immersion could make users more susceptible to malicious push notifications. This risk could impact users at both the individual and organization levels as well as across industries, particularly as smart wearable devices become increasingly immersive. Stage 1 of this qualitative empirical study used a virtual presentation to simulate the user interfaces of the popular AR applications Google Lens, Google Translate, Instagram, Maps, and Pokémon GO presented to ($N = 70$) participants aged 18–40 who regularly used these applications. Of the two notification themes presented – familiarity and urgency – 62 of 70 participants chose the familiarity theme with which to engage. Based on these results, stage 2 of the study consulted four experts in the field of AR application development to design an artificial intelligence-equipped feature that could intercept possibly malicious artifacts entering the user's line of sight during partial immersion in AR. This article proposes the design for a natively embedded security application configurable across all device operating system types to assess incoming content in real time. The article then draws upon the expertise of these four participants to inform a comparative analysis assessing how the heuristic sentiment analytic algorithm used by such an application compares against existing spam filter algorithms. The greatest advantage found in heuristic sentiment analysis that would improve upon existing spam filter techniques, such as Bayesian and rule-based detection, was the decreased reliance on user input. The proposed automated tool's combination of heuristic threat recognition of emerging threats and sentiment analysis based on a pre-configured lexicon could reduce the overall time required to intercept malicious content incoming to a smart wearable device interface.

Keywords: application development, artificial intelligence, cyberpsychology, cybersecurity, smart wearable technologies

1. Introduction

Cyber threat actors often use social engineering attacks to lure unsuspecting individuals and organizations to disclose sensitive information. This risk affects many industries, including critical sectors like aerospace operations and healthcare.

Provided the heightened risk of distraction during interactive immersion like augmented reality (AR) and virtual reality (VR) adapted for a narrower interface such as with a smart wearable device like an Apple Watch, users might inadvertently engage with malicious content that suddenly enters the user interface (UI), either by clicking or swiping to remove the content or because the content projects a sense of familiarity by impersonating someone known to the user [1]. Popular themes used in social engineering attacks are familiarity and urgency, with familiarity being particularly persuasive due to the sense of trust invoked [2].

While spam filters have long been used in an email context, fewer solutions operate by intercepting suspicious content entering the UI in real time, particularly for wearable devices. Therefore, the various traditionally used analysis techniques like Bayesian and rule-based detection tend to operate on an individual basis rather than together and also often rely upon user input. Thus, this decentralized detection framework and reliance on user judgment could risk malicious content evading detection. An automated solution could use heuristic sentiment analysis or the combination of artificial intelligence (AI)-supported threat assessment (heuristic threat analysis) with AI detection typically used for social media content filtering (sentiment analysis) [3, 4].

2. Literature Review

Despite thorough research on social engineering and AI threat detection as separate disciplines, a research gap appears in terms of how AI can improve malicious content detection, particularly in a real-time context, to better protect users of interactive and wearable technologies.

Schmitt and Flechais discussed how AI could be used to improve upon existing email spam filter parameters such as Bayesian, blacklist, content filtering, header, language, and rule-based by analyzing both social engineering tactics used and security application mitigation [5].

*Corresponding author: Sarah Katz, Department of Cyberpsychology, Capitol Technology University, USA. Email: skatz@captechu.edu

On the topic of static versus dynamic analysis, Jáñez-Martino et al. [6] presented potential issues that emerge when spam filter machine learning (ML) algorithms are not updated to include evolving threat actor tactics. Yet, this critical analysis highlights the risks more than the solutions to such a challenge.

Therefore, a solution able to identify dynamic threat actor tactics by combining threat detection with real-time content monitoring could help close these gaps.

Although existing studies have explored the security risks of extended reality and the threat of social engineering, a research gap also arises surrounding push notifications as a tertiary factor in the distraction of an immersed user as well as how such threats can be mitigated via automation. In fact, sentiment analysis has more often been utilized for social media content monitoring rather than for social engineering mitigation.

The review below includes existing literature on the risks and utility of extended reality (AR and VR) as well as how AI-based automation is currently used for areas of risk mitigation such as with social media monitoring. Developers could use this information alongside research into social engineering susceptibility due to decreased cognitive processing capacity as a foundation for a tool that leverages heuristic documentation to analyze suspicious content in the place of the human user in a wearable device interface.

2.1. The impact of interactive media on cognition

For a holistic standpoint on the role of extended reality for today's society, the information systems auditing organization ISACA discussed various applications of augmented and virtual reality in the workplace. These technologies are highly leveraged for video gaming and training across many industries [7]. The disadvantages of interactive media use in the workplace include the risk of software vulnerabilities and privacy infringement, particularly from developer-side user data gathering.

In particular, Rajan et al. [8] found that wearable technologies have been known to cause significant distraction levels due to barrages of push notifications, such as with a smart watch.

While Kohnke [7] reviewed the advantages and shortcomings of immersive technologies in the workplace, the discussion does not suggest a definitive solution. In this case, the risk of threat actors using an immersive application to exploit user distraction for social engineering is not explored.

On the other hand, Turner [9] discussed the risk of distraction and deception during AR immersion, including susceptibility to scams following engagement with malicious onscreen virtual objects and incidents such as a car accident due to a lack of attention toward real-world surroundings. However, this source did not offer any in-app or native operating system (OS) automated solutions to counter these risks. The article also did not examine how common social engineering-specific message themes such as familiarity and urgency might influence the user's tendency to engage with the content.

Meanwhile, Ferreira and Teles [10] showcased the use of persuasive titles in phishing emails and the subsequent usefulness of automated mitigation to help detect this type of social engineering. While social engineering and automated protection were both discussed, this study did not touch on the backend design of such automation tools. Moreover, the additional distraction of new objects entering a user's line of sight during immersion is not discussed.

Still, the information on various lure themes such as authority and familiarity could help develop a lexicon used by an automated detection tool for flagging suspicious content.

Although Giaretta [11] reviewed privacy and security concerns with AR and VR, they did not assess the risks of an onscreen notification suddenly entering the UI. As in-app deception of this sort often requires prior attacker access to the application, investigating for any pre-compromise activity could be helpful.

Regarding interactive media use with smart wearable technologies in particular, while Stasolla [12] presented compelling evidence for the medical benefits of this combination, this study did not explore the potential risks involved, such as distraction.

Strecker et al. [13] recommended optical character recognition for text and image analysis entering a UI during immersion in an extended reality interface. This form of object recognition basically parallels a hypothetical heuristic-based sentiment analysis method of detecting incoming content to help unburden the user's cognitive processing capacity. Given the relatively small UI of many wearable devices, reducing cognitive load is essential for safeguarding user well-being and security from cyber threats.

A more detailed examination of sentiment analysis for social engineering mitigation follows in the next section.

2.2. Sentiment analysis and the psychology of social engineering

In their review, Xu et al. [14] discussed lexicon-based versus ML methods of sentiment analysis – that is, the pre-defined terminology contrasted with the more dynamic ML capabilities for AI content assessment. This research examined how sentiment analysis in particular has primarily served for social media content moderation.

Even though this study did not look into specific uses for social engineering in immersive environments, the proposed automation tool addressed in the present study would benefit from exploration into how the lexicon and ML components of sentiment analysis could be merged.

While the Federal Communications Commission [15] issued a centralized guide on whether to respond to a call, this advice does not necessarily translate to push notifications or specific themes used by threat actors beyond the scenario when a call does not originate from a local caller.

That said, sentiment analysis, historically leveraged to assess threatening and other undesirable content on social media, could be employed to assess for high-risk content in push notifications and other pop-ups. While Kenny et al. [16] emphasized social media content analysis rather than push notifications, the same general parameters could apply for analyzing sudden onscreen notifications within an immersive interface.

As an early study on malicious push notifications, Hyun et al. [17] looked into malware delivery instead of social engineering and did not consider the UI at hand (immersive interface vs. regular mobile vs. desktop).

Still, information about the latest prominent malware is also relevant for when a user unintentionally interacts with a malicious push notification by unwittingly clicking a link that installs malware on the user's device.

Meanwhile, Kaur et al. [18] cited sentiment analysis as helpful AI approach to information security, alongside others such as text mining for deeper analysis of keywords in message content, image processing for image-based threat monitoring, and large language models as well as natural language processing (NLP) for deciphering potentially malicious content in multiple languages. All of these technologies could be implemented into a security feature to investigate incoming content before the user has an opportunity to engage with the message in question.

While this study did not touch upon social engineering mitigation specifically, the roles of these different AI technologies for such mitigation remain essential. Used in combination, heuristic sentiment analysis could determine content threat level based not only on a pre-determined lexicon of suspicious key terms and a database of emerging threats but also AI consideration of whether the content is recognized or expected.

On the whole, plentiful research exists into social engineering and the benefits as well as risks of immersive technologies. However, a gap remains as to how distraction stemming from extended reality immersion could be exploited by sudden pop-up messages and other objects entering the UI of a smart wearable device, as pictured in Figure 1 [19]. Furthermore, few studies have explored how AI-supported sentiment analysis could be blended with heuristic threat detection to help mitigate these challenges.

The gaps found for each traditional spam filter technique is thusly presented with its corresponding parameter in the proposed solution, based on the objective to perform threat analysis that draws from both existing user input, such as device-recognized and trusted message origins, and real-time external research, such as open-source intelligence (OSINT) collection from legitimate external sources.

Figure 1
A missed call notification displayed on a smart watch UI with a notification

3. Research Methodology

3.1. Research design

The first stage of this research involved a simulation of three identical AR UI interfaces, from which 70 regular AR users aged 18–40 were asked to select one of two animated push notifications entering the UI – (A) familiarity, "A friend is calling in," and (B) urgency, "Update or device will restart in 5 minutes." Participants were then asked to explain their choice rationale for which collated responses showed a trend toward social pressure to respond to a connection.

Drawing upon the majority tendency to choose familiarity as seen in Table 1, content with this theme could be implemented first into the tool's detection schema as a starting point.

Table 1
Familiarity versus urgency theme

Familiarity	62
Urgency	8

For simplicity's sake, based on research by Adamowicz et al. [20] showing Python language as preferable for data collection and analysis in smart wearable technology, the proposed security application will be referred to as QuikKatch. The tool would use Python code as a foundation for analyzing an individual user's tendency to engage with an onscreen notification denoting a certain lure theme. The application would then assess the threat level of incoming content compared to the user's likelihood to interact with the themes presented.

While not yet validated through user testing, the framework for this proposed tool was designed based on a consultation with four experts from the fields of AI engineering, information security, and user experience.

3.2. Participants

Alongside the 70 participants from stage 1, stage 2 participants were selected based on their fields of expertise and asked the following questions via email communication:

1) Would it be possible to implement a mobile application with a computer vision-equipped feature capable of detecting when a push notification or pop-up appears on screen during application use? Further, could such a feature analyze only a pop-up that came in through the application itself or also a pop-up that simply entered the UI while the application was in use?
2) If the feature mentioned in question #1 could be feasible, what would be the mechanics behind its backend development? That is, would it need to be synced to the native OS, or could it detect any pop-up/push notification that enters the application's UI while the application is open?
3) Could such a feature analyze the content of the push notification or pop-up?
4) If the feature were equipped to analyze the pop-up content, could it perhaps use AI to distinguish potentially harmful from benign content and notify the user in real time?

5) Given any limitations of the proposed feature above, which of the aforementioned functionalities would you imagine could be achieved?

3.2.1. Technical details

The second interview explored a more detailed design of this detection feature with the following question:

> What could be a manner in which a mobile computer vision feature would integrate with system notification APIs to analyze and intercept certain suspicious notifications in real-time (the objective being to prevent the user from automatically swiping to make the notification disappear)?

3.2.2. Potential challenges for user experiences

The fourth interview posed the following question to examine possible challenges related to the in-use AR application that would hypothetically be open and running at the same time as the detection application:

> As a UX specialist, which obstacles could you personally imagine arising from a computer vision application potentially conflicting with an app of yours that is running on a device (e.g., user view obfuscation)? What test methods might be possible to help mitigate these obstacles?

3.2.3. Instruments

Based on the participants' responses, Table 2 was illustrated to demonstrate a comparative analysis between existing popular spam filters and which improvements the proposed tool QuikKatch would aim to offer.

By and large, the proposed heuristic sentiment analysis algorithm would expand upon traditional spam filter algorithms by supplementing pre-defined suspicious terms and unrecognized incoming communication sources with the latest threats reported by reputable global cybersecurity and news platforms.

In summary, the above table was informed by the expert responses to how such an automated tool would need to function on both a pre-defined sentiment analysis and a more heuristic level for real-time threat detection. While this application likely wouldn't cause any physical user discomfort, a challenge was cautioned involving user privacy concerns surrounding an application with screen capture capabilities.

Table 2
Traditional spam filters versus QuikKatch

Traditional filter parameter	Efficiency gap	QuikKatch
Bayesian	Relies on user input supplied over time	Uses heuristic threat analysis that uses user input as one component of, rather than the sole determinant of, content risk level
Blacklist	Relies on pre-determined lexicon, known as a "blacklist"	Supplements user-supplied parameters with new threats, based on keywords related to emerging threat actors and malware types gathered from reputable platforms
Content filtering	Relies on pre-determined lexicon provided either by the user or a governing body such as a cybersecurity agency or both	Supplements user/agency-supplied lexicons of threatening material with an evolving lexicon based on emerging threat actors and malware types, including recently discovered URLs and code provided by cybersecurity and news agencies
Header	Bases legitimacy on whether the sender and reply-to headers match	In instances of such discrepancy, searches for a historical legitimate association between the recipient and content sender to help reduce false positives
Language	Might not account for non-English content	Ensures analysis capabilities of as many world languages as possible
Rule-based	Relies on a rule set that might not be continually updated	Assesses using both individual device user and reputable security organization data protection standards (e.g., OSCE for Europe and NSA for the United States)

4. Conclusion

As users face increasingly distracting interactive experiences such as AR and VR, particularly within smaller UIs such as with smart wearable devices, evolving mitigation measures against social engineering and other human factor-dependent cyberattacks remain paramount. Thus far, sentiment analysis and heuristic threat detection have not been widely leveraged in unison to assess threats in real time, particularly social engineering attacks against users of smart wearable technology. An automated solution like QuikKatch could help widen the protective scope of existing content filters by merging dynamic, non-user-dependent heuristic threat detection with nuanced sentiment analysis in assessments of real-time incoming content. Such automated analysis could further protect against emerging as well as recognized threats.

Given this risk spans multiple disciplines, this tool could benefit many industries by approaching the issue from a standpoint that combines cybersecurity, cyberpsychology, and application development, applied specifically to the relatively still-emerging space of interactive technologies like AR and VR in the context of wearable technologies such as the smart watch.

Lastly, such a tool using heuristic sentiment analysis for threat detection could prove useful to individual users and organizations

alike. From the latter perspective in particular, CISO-level executives as well as security practitioners focusing on smart wearable devices could collaborate in realizing this defensive technology.

In an academic context, AI researchers and engineers specializing in NLP and computer vision could explore how threat heuristics and pre-determined databases for sentiment analysis could inform one another in providing revolutionary real-time protection for all varieties of interactive media users utilizing smart wearable technologies.

Recommendations

The above analysis illustrates an overreliance on user input among traditional spam filters for both content analysis and threat detection. This gap risks the Bayesian and rule-based techniques resulting in outdated or static threat databases, which calls for dynamic heuristic analysis to bolster the sentiment analysis often used in assessing the threat level of incoming content more generally.

As such, these traditional filters tend not to draw many resources from news sources and reputable security organizations. QuikKatch would aim to enhance social engineering mitigation by supplementing user awareness with automated tracking of evolving threats potentially unaccounted for by the human user, prioritizing content with familiarity and urgency lures entering the wearable device interface.

This tool would leverage dynamic heuristics by analyzing in real time all newly cited threats covered by both international security agencies and reliable media. This analysis technique would add to existing blacklists and content filters, including any named threat actors and malware types that might appear in incoming notifications to the wearable device UI.

Finally, the tool would adapt the header analysis technique typically used for examining email sender headers to investigate the origin of incoming notifications. If the content comes in the form of a text message or push notification that originates from a phone number, domain, or IP address not normally contacted by the device at hand or associated with known malicious activity, as documented by sources such as VirusTotal, QuikKatch would intercept the message.

Figure 2 demonstrates an overview of the proposed tool design.

The following Python script provides a sample method combining heuristic threat detection and sentiment analysis to analyze the texts and image content of incoming messages and notifications. The sentiment analysis uses a pre-defined lexicon of common threatening terms, while the heuristic component searches reputable security agencies and news platforms for recently discovered threats. For the purposes of this study, the sample security organization is the National Security Agency, and the sample news platform is BBC News.

The code in Figure 3 includes both adherence to a pre-defined lexicon installed within the program and the search functionality to identify the latest threat stories from reputable security agencies and news platforms.

The code illustrates a combination of traditional email filter techniques, thus helping to cover multiple bases of threat protection all at once. This coverage includes supplementing reliance on the human user with external resources to support a dynamic heuristic database of new as well as known threats in any incoming content to the user's device, particularly while an interactive application is also running on said device.

The code parameters operate according to (1) a pre-defined lexicon pre-installed into the application and (2) a functionality to call on all most recent content provided by international cybersecurity organizations and reputable news outlets. In this way, the first parameter addresses the sentiment analysis component, while the second parameter expands the application's scope to include emerging threats.

In the event that a 204 or similar error occurs, the application could inform the user via an error message with a notification of automatic reattempt every 5 minutes until success.

The tool would also analyze content in multiple global languages, thus reducing the threat surface when it comes to potential geopolitical threats such as attacks by nation-state actors that use social engineering to deceive distracted targets.

Finally, any user data privacy concerns would need to be addressed by a compliance and legal consultation entity.

The code for analyzing the user's likelihood to engage and subsequent interception of an incoming notification deemed suspicious follows in Figure 4.

Limitations and Future Research

The main limitation with the proposed technology lies with how much data and therefore threat exposure the application might need to gather from the user's wearable device activity before learning the

Figure 2
Proposed QuikKatch design

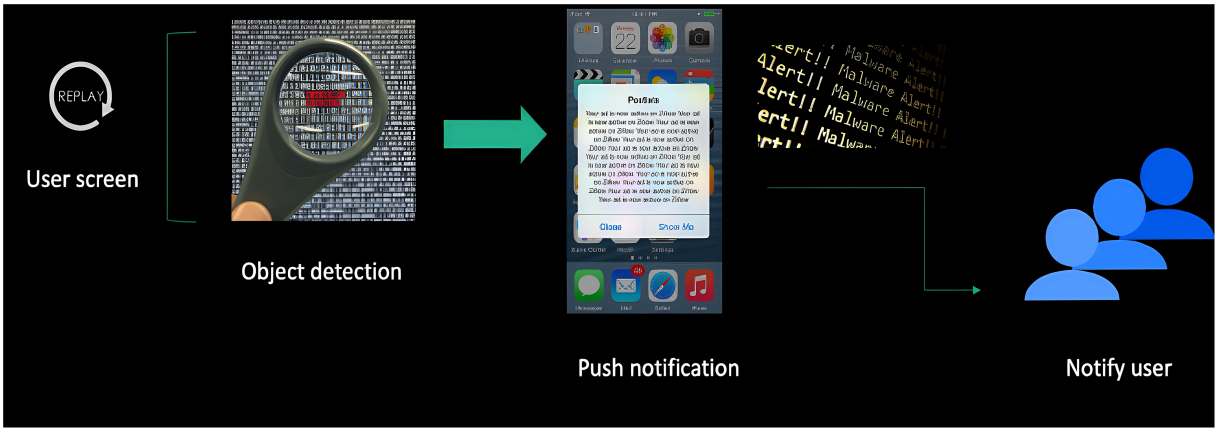

Figure 3
QuikKatch Python script: pre-defined lexicon adherence and incoming content threat level analysis

```
import re
from urllib.parse import urlparse

# Predefined threatening terms (example terms)
THREAT_LEXICON = {
    "attack", "bomb", "kill", "shoot", "terror", "explosive", "assassinate",
"threat"
}

# Malicious URLs reported (example entries)
MALICIOUS_URLS = {
    "malicious-site.com", "phishing.example.org", "badurl.bbc-warning.net",
"dangerous.nsa.gov"
}

# Simulated list of push notifications
push_notifications = [
    "Check this out: http://malicious-site.com, it's important!",
    "There might be a bomb at the station!",
    "Friendly reminder to stay safe :)",
    "Don't visit phishing.example.org, it's a trap!",
    "The parade might face a threat today.",
    "This link is totally safe: http://example.com"
]

def extract_urls(text):
    """Extracts all URLs from the given text using regex."""
    url_pattern = re.compile(r'https?://[^\s]+')
    return url_pattern.findall(text)

def is_threatening(message):
    """Check if the message contains any threatening terms."""
    words = set(re.findall(r'\b\w+\b', message.lower()))
    return THREAT_LEXICON.intersection(words)

def contains_malicious_url(message):
    """Check if the message contains any malicious URLs."""
    urls = extract_urls(message)
    for url in urls:
        domain = urlparse(url).netloc.lower()
        if any(malicious_domain in domain for malicious_domain in
MALICIOUS_URLS):
            return True
    return False

def analyze_notifications(notifications):
    """Analyze a list of notifications for threats and malicious content."""
    for i, message in enumerate(notifications, 1):
        print(f"Analyzing message {i}: {message}")
        threats = is_threatening(message)
        malicious = contains_malicious_url(message)

        if threats or malicious:
            print("⚠ Threat detected:")
            if threats:
                print(f" - Threatening terms: {', '.join(threats)}")
            if malicious:
                print(" - Malicious URL detected")
        else:
            print("✓ No threats detected.")
        print("-" * 60)

# Run analysis
analyze_notifications(push_notifications)
```

Figure 4
QuikKatch Python script: data gathering based on user tendency to interact with a notification denoting familiarity versus urgency

```
import sqlite3
import re
import requests

# URL for the CISA malicious domains list (assumed)
CISA_MALICIOUS_DOMAINS_URL =
"https://www.cisa.gov/malicious-domains-list"

# Initialize SQLite database
DB_NAME = "push_notifications.db"

# Create tables for tracking engagement
def initialize_db():
    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
    cursor.execute("""
        CREATE TABLE IF NOT EXISTS notifications (
            id INTEGER PRIMARY KEY AUTOINCREMENT,
            message TEXT,
            category TEXT CHECK(category IN ('Familiarity',
'Urgency')),
            engagement TEXT CHECK(engagement IN ('Clicked',
'Dismissed', 'Ignored'))
        )
    """)
    conn.commit()
    conn.close()

# Fetch latest CISA malicious domains list
def get_malicious_domains():
    try:
        response = requests.get(CISA_MALICIOUS_DOMAINS_URL)
        response.raise_for_status()
        return set(response.text.split("\n"))
    except requests.RequestException:
        print("Failed to fetch malicious domains. Using empty list.")
        return set()

# Function to check if a notification contains a malicious URL
def contains_malicious_url(message, malicious_domains):
    urls = re.findall(r"https?://[^\s]+", message)  # Extract URLs
    for url in urls:
        for domain in malicious_domains:
            if domain in url:
                return True
    return False

# Log push notification engagement
def log_notification(message, category, engagement,
malicious_domains):
    if contains_malicious_url(message, malicious_domains):
        print(f"Blocked malicious notification: {message}")
        return  # Do not log blocked notifications

    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
    cursor.execute("INSERT INTO notifications (message, category,
engagement) VALUES (?, ?, ?)",
                   (message, category, engagement))
    conn.commit()
    conn.close()
    print(f"Logged: [{category}] {message} - Engagement:
{engagement}")

# Get basic analytics on engagement
def analyze_engagement():
    conn = sqlite3.connect(DB_NAME)
    cursor = conn.cursor()
```

individual's habits well enough to allow or block incoming notifications. Therefore, databases should be expanded based on both existing lexicons and up-to-date external resource analyses that train on familiarity as well as urgency and less persuasive lure themes.

The code would also need to be significantly adapted to include other programming languages, such as Kotlin for Android and Swift for iPhone, for integration with specific types of mobile OSs.

Finally, privacy concerns such as user data gathering by the device would also have to be considered.

Acknowledgment

The author is grateful for instruction under the faculty at Capitol Technology University.

Conflicts of Interest

The author declares that she has no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Sarah Katz: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] Gronowski, A., Arness, D. C., Ng, J., Qu, Z., Lau, C. W., Catchpoole, D., & Nguyen, Q. V. (2024). The impact of virtual and augmented reality on presence, user experience and performance of information visualisation. *Virtual Reality*, *28*(3), 133. https://doi.org/10.1007/s10055-024-01032-w

[2] Katz, S. (2024). Leveraging artificial intelligence to mitigate user susceptibility to malicious push notifications during augmented reality immersion. *Annual Review of Cyber-therapy and Telemedicine*, *22*, 232–237.

[3] Jayaprakash, R., Natarajan, K., Daniel, J. A., Chinnappan, C. V., Giri, J., Qin, H., & Mallik, S. (2024). Heuristic machine learning approaches for identifying phishing threats across web and email platforms. *Frontiers in Artificial Intelligence*, *7*, 1414122. https://doi.org/10.3389/frai.2024.1414122

[4] Jain, R., Kumar, A., Nayyar, A., Dewan, K., Garg, R., Raman, S., & Ganguly, S. (2023). Explaining sentiment analysis results on social media texts through visualization. *Multimedia Tools and Applications*, *82*(15), 22613–22629. https://doi.org/10.1007/s11042-023-14432-y

[5] Schmitt, M., & Flechais, I. (2024). Digital deception: Generative artificial intelligence in social engineering and phishing. *Artificial Intelligence Review*, *57*(12), 324. https://doi.org/10.1007/s10462-024-10973-2

[6] Jáñez-Martino, F., Alaiz-Rodríguez, R., González-Castro, V., Fidalgo, E., & Alegre, E. (2023). A review of spam email detection: Analysis of spammer strategies and the dataset shift problem. *Artificial Intelligence Review*, *56*(2), 1145–1173. https://doi.org/10.1007/s10462-022-10195-4

[7] Kohnke, A. (2020). The risk and rewards of enterprise use of augmented reality and virtual reality. *ISACA Journal*, *1*, 16–23.

[8] Rajan, Y. P., Aiswarya, B., & Selvi, A. J. A. (2025). Disruptive charms of wearable technologies: Navigating digital distractions and work performance. *Journal of Information Technology Teaching Cases, 15*(1), 53–58. https://doi.org/10.1177/20438869231203341

[9] Turner, C. (2022). Augmented reality, augmented epistemology, and the real-world web. *Philosophy & Technology*, *35*(1), 19. https://doi.org/10.1007/s13347-022-00496-5

[10] Ferreira, A., & Teles, S. (2019). Persuasion: How phishing emails can influence users and bypass security measures. *International Journal of Human-Computer Studies*, *125*, 19–31. https://doi.org/10.1016/j.ijhcs.2018.12.004

[11] Giaretta, A. (2025). Security and privacy in virtual reality: A literature survey. *Virtual Reality 29*(1), 10. https://doi.org/10.1007/s10055-024-01079-9

[12] Stasolla, F. (2021). Virtual reality and wearable technologies to support adaptive responding of children and adolescents with neurodevelopmental disorders: A critical comment and new perspectives. *Frontiers in Psychology*, *12*, 720626. https://doi.org/10.3389/fpsyg.2021.720626

[13] Strecker, J., García, K., Bektaş, K., Mayer, S., & Ramanathan, G. (2023). SOCRAR: Semantic OCR through augmented reality. In *Proceedings of the 12th International Conference on the Internet of Things*, 25–32. https://doi.org/10.1145/3567445.3567453

[14] Xu, M., Luo, Y., Zhang, Y., Xia, R., Qian, H., & Zou, X. (2023). Game-based learning in medical education. *Frontiers in Public Health*, *11*, 1113682. https://doi.org/10.3389/fpubh.2023.1113682

[15] Federal Communications Commission. (2024). *Stop unwanted robocalls and texts*. Retrieved from: https://www.fcc.gov/consumers/guides/stop-unwanted-robocalls-and-texts

[16] Kenny, M., Pitropakis, N., Sayeed, S., Chrysoulas, C., & Mylonas, A. (2024). Malicious insider threat detection using sentiment analysis of social media topics. In *ICT Systems Security and Privacy Protection: 39th IFIP International Conference*, 264–278. https://doi.org/10.1007/978-3-031-65175-5_19

[17] Hyun, S., Cho, J., Cho, G., & Kim, H. (2018). Design and analysis of push notification-based malware on Android. *Security and Communication Networks*, *2018*(1), 8510256. https://doi.org/10.1155/2018/8510256

[18] Kaur, R., Gabrijelčič, D., & Klobučar, T. (2023). Artificial intelligence for cybersecurity: Literature review and future research directions. *Information Fusion*, *97*, 101804. https://doi.org/10.1016/j.inffus.2023.101804

[19] Apple. (2024). *Notifications on your Apple Watch*. Retrieved from: https://support.apple.com/en-us/108369

[20] Adamowicz, L., Christakis, Y., Czech, M. D., & Adamusiak, T. (2022). SciKit digital health: Python package for streamlined wearable inertial sensor data processing. *JMIR mHealth and uHealth*, *10*(4), e36762. https://doi.org/10.2196/36762

How to Cite: Katz, S. (2025). Heuristic Sentiment Analysis for Social Engineering Mitigation During Interactive Immersion with Smart Wearable Technology. *Smart Wearable Technology*, *1*, A10. https://doi.org/10.47852/bonviewSWT52025469