CAVA: Cognitive Aid for Vulnerability Analysis

Evelyn Kim[1], Sunny J. Fugate[2], Christian Lebiere[3], Aidan Barbieux[1], Jonathan M. Buch[2], Jaehoon Choe[1], Edward A. Cranford[3], Joseph DiVita[2], Jeremy P. Johnson[2], Mia Levy[1], Froylan Maldonado[2], Brianna Marsh[1], Donald Morrison[3], Jocelyn Rego[1], Mitchell Sayer[1], Alex Waagen[1], Rajan Bhattacharyya[1]

[1]HRL Laboratories [2]Naval Information Warfare Center, Pacific [3]Carnegie Mellon University
ekim@hrl.com

Abstract

Becoming a reverse engineer (RE) requires rigorous training and understanding of program structure and functionality, and experts develop heuristic strategies and intuitions from real-world experiences. This paper attempts to capture REs' strategies and intuitions within a predictive cognitive model and demonstrate the feasibility of assisting novice REs using an intelligent recommender called CAVA (Cognitive Aid for Vulnerability Analysis). CAVA leverages physiological sensors to assess a novice's cognitive states and provides real-time visual hints when the novice's attention and engagement diminish. We instrumented Ghidra and conducted pilot experiments with REs. Open-loop experiments with 9 REs confirmed the feasibility of identifying novices from experts using physiological signals, and a pilot closed-loop experiment tested the feasibility of providing visual recommendations to a novice. Despite challenges in recruiting REs, our progress suggests that CAVA is a promising approach to improve novice performance and our understanding of experts' behavior when performing complex real-world reverse engineering tasks.

1. Introduction

Understanding how cyber attackers exploit software vulnerabilities is non-trivial, even for experienced analysts. Defending against such an attack requires writing a code without vulnerabilities or finding them *before* practical exploitation. Unfortunately, reverse-engineering a code is a daunting task.

In fact, attackers exploit previously vulnerabilities that have not been fully patched, indicating that defenders cannot catch up with attackers, according to Google Project Zero[1]. In 2022, 41 zero-day vulnerabilities have been exploited, and 40% of them were simply variants of previously-patched bugs, with more than 20% being variants of previous 2021 in-the-wild zero-day bugs. This trend follows what Google Project Zero observed in 2020, where 25% of all zero-day vulnerabilities were variants of previously disclosed vulnerabilities. This finding highlights the downside of either automated machine learning (ML)-based automation tools that suffer from non-zero false alarms [12, 6] or systems that dynamically diversity or obfuscate code to delay attacks [1], and how human performance could bridge the gap. In other words, 25-40% of zero-day exploits could have been avoided if thorough investigation was conducted with human reverse engineers with proper tools.

Training can help reverse engineers (REs) understand common vulnerability types and exploitation methods to some limit, because REs require good intuitions and strategies that experts develop from real-world experiences. Online communities and forums provide opportunities for novice REs to get support from other REs and gain insights [23]. However, a major limitation is receiving support in a timely manner. An ideal approach to train a novice RE includes an expert RE who provides inputs specific to the vulnerability cases, but such a training approach is expensive and not scalable.

This paper attempts to capture expert REs' intuitions and strategies in a cognitive model, and visually provide them to novices as an intelligent recommender. We propose a closed-loop recommender called CAVA (Cognitive Aid for Vulnerability Analysis) that leverages on-body and off-body physiological sensors to assess a novice RE's cognitive states, such that the recommender chimes in with hints when the novice's attention and engagement diminishes in real time. Hints are provided by CAVA's cognitive model that is trained with experts' strategies. In concert, when a novice's cognitive load is high and the projected behavior deviates from that of experts, the cognitive model provides a visual recommendation using different hues and lightness.

We instrumented the Ghidra Reverse Engineering Framework, and developed a deep learning approach to extract the specifics of the behavioral data, including keyboard and mouse activities, at different panels. We also incorporated the cognitive visualizer in Ghidra.

[1]https://security.googleblog.com/2023/07/the-ups-and-downs-of-0-days-year-in.html

URI: https://hdl.handle.net/10125/107272
978-0-9981331-7-1

To evaluate the effectiveness of CAVA, we conducted pilot experiments with REs with varying levels of experience and skillset. The open-loop CAVA evaluation tested the feasibility of identifying novices from experts using behavioral and physiological data, and the closed-loop CAVA evaluation tested its effectiveness and usability. In total, we recruited 9 REs for the open-loop evaluation, and 1 for the closed-loop evaluation. Our results hint that CAVA is a promising approach to virtually support novices to become experts in reverse engineering with trust and usability without incurring additional cognitive load.

Contributions. CAVA is the a closed-loop intelligent recommender that provides expert REs' intuitions and strategies when a novice RE encounters challenges in exploring and exploiting a potential vulnerability:

- CAVA utilizes a cognitive model that estimates the novice's cognitive states based on the neurophysiological and behavioral responses.
- CAVA visualizes experts' recommendations when the novice experiences stress, fatigue, and overload.
- CAVA is the closed-loop system that has been integrated with Ghidra V9.0 with neurocognitive analysis pipeline, cognitive model, and cognitive visualization.
- CAVA has shown promising results from pilot experiments with REs: 38% utilization of recommendations, reduced user confusion, reduced time on vulnerability analysis, and high usefulness. CAVA has potential to transfer experts' strategies.

2. Problem Definition

This paper attempts to (1) understand how expert reverse engineers (REs) navigate a piece of code until they identify points of interest (POIs) and points of vulnerabilities (POVs), and (2) apply (1) to provide visual recommendations to novices when they are cognitively overloaded. Eventually, visual recommendations can support novices to learn experts' strategies and apply insights to other reverse engineering tasks.

2.1. Desired Properties

- **Engagement & trust:** CAVA should not reduce the level of engagement as an RE receives visual recommendations, and an RE should not lose trust in CAVA's recommendations.
- **Computational and communication efficiency:** Multiple components are involved in CAVA, such as (1) measuring physiological states using sensors, (2) inferring cognitive states to determine the visual recommendation types, and (3) displaying recommendations. All steps should not incur computational and communication overhead to ensure that an RE receives recommendations as needed.

Table 1. List of physiological sensors. * indicates on-body sensors.

Sensors	Functionalities
32-channel actiCHamp EEG*	Scalp electrode array measuring confluences of brain activity
Portalite fNIRS*	Uses blood oxygenation for metabolic activity within specific brain areas
Shimmer GSR+*	Measures the electrical resistance of the skin
Shimmer HR*	Measures heart rate that corresponds to emotional arousal
Smarteye AIX eyetracker	Camera calculates eye gaze, head position, pupil size, and correlates pupil dilation to objects of interest

2.2. Assumptions

CAVA utilizes on-body and off-body sensors to collect physiological responses and infer behavioral and cognitive states (Table 1). Some sensors require active calibrations, requiring REs to follow instructions (e.g., eyetracker, EEG), while other sensors can calibrate passively by observing signals. In our experiment (Section 5), sensor placement and calibration took approximately 30-45 minutes. We assume that on-body sensor placement and calibration time do not impact the quality of data that CAVA collects for its closed-loop recommendation generation.

3. CAVA System Overview

When a novice RE explores a piece of software using a vulnerability analysis tool, CAVA's primary objective is to guide the novice with the next action as if an expert RE is with the novice, especially when the novice seems to be cognitively overloaded (e.g., degraded attention, fatigue, etc.). CAVA attempts to (1) capture expert REs' behavior, (2) estimate the novice's cognitive state using a variety of physiological sensors as mentioned in 1 along with behavioral movement from the keyboard and mouse, and (3) provide hints in a timely manner without increasing the novice's cognitive workload. CAVA uses Ghidra, which is an open-source software reverse engineering tool developed by National Security Agency [2].

Example. Alice, who is a novice RE, is currently examining a program with a potential SQL injection vulnerability using a reverse engineering tool (Figure 1).

[2] https://ghidra-sre.org/

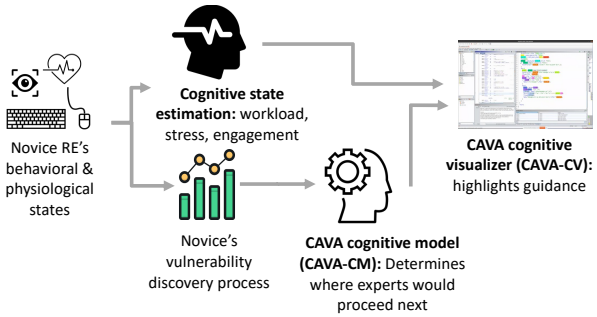

Figure 1. CAVA overview. Given previously-captured expert REs' strategies, CAVA estimates the novice's cognitive state using physiological sensors, keyboard, and mouse activities, and provides hints in a timely manner without increasing cognitive workload.

Alice is also using CAVA that constantly measures and infers her cognitive states as she performs vulnerability analysis. The code is quite complex with many reference pointers, and CAVA detects that Alice might feel stressful and overwhelmed. To ensure that Alice remains focused to finish the current task, CAVA's cognitive visualizer (CAVA-CV) gets inputs from cognitive model (CAVA-CM) to highlight lines in the program as follows: (1) CAVA-CM notifies CAVA-CV to highlight the previous n lines of code that Alice investigated to offload them from her memory. The intensity of the highlights is consistent with the memory decay (i.e., recently-visited lines are more vivid compared to others). (2) Based on previously-collected experts' strategies, CAVA-CM informs CAVA-CV to use another color to highlight the next line that Alice should examine.

4. Cognitive Recommender System

Modeling an expert RE's mental and cognitive states is crucial in developing a cognitive recommender. In this section, we describe details of cognitive model (CAVA-CM) in (1) capturing experts' strategies, (2) predicting novices' actions, and (3) providing visual recommendations (CAVA-CV).

4.1. CAVA Cognitive Model (CAVA-CM)

CAVA uses Adaptive Control of Thought-Rational (ACT-R), which is a cognitive architecture for creating a wide variety of highly accurate models of human cognition.[3] ACT-R provides a theoretical framework to model complex human cognition and processes such as memory retrieval, pattern matching, and decision making [2]. ACT-R integrates symbolic knowledge and sub-symbolic computations to reflect individual differences and emergent cognitive biases, and accounts for training effects by modeling learning processes [14]. Using ACT-R, we developed a Generalized Decision Making (GDM) model to represent the mental models of Ghidra users. The GDM model personalizes the cognitive model against an individual or groups of individuals.

[3] https://act-r.psy.cmu.edu/publication/

4.2. CAVA Cognitive Visualizer (CAVA-CV)

A cognitive visualizer (CAVA-CV) represents and attempts to predict cognitive activity in two-fold: (1) augment limited cognitive resources such as working memory by representing recent activity in a way that support current activities; (2) predict future activity to provide recommendations over possible actions to facilitate decision making and scaffold the development and transfer of expertise. The Instance-Based Learning (IBL) models are at the core of CAVA-CV's decisions to readily use the experimental behavior and align model behavior against the experimental trace. IBL models' predictions on future decision and behaviors can be used to optimize interventions such as dynamic layout recommendations and cognitive visualizations.

As users navigate the code, each action (e.g., mouse clicks) is represented as an experience in memory. After a user takes an action, CAVA-CV leverages activation computations to represent cognitive activity in visualizations such as code highlighting. Activation is composed of a number of additive factors, combined using a Bayesian framework. Recency represents the near-term working memory context for local situation awareness (e.g., by helping to remove accidental returns to already examined pieces of code). Frequency represents the longer-term distribution of activity, emphasizing importance and centrality (e.g., frequency of visiting some pieces of code indicates important information). Association represents sequential patterns of activity capturing structural dependencies such as function calls or control structures (e.g., providing a natural exploration path over the call graph). Similarity represents semantic features in code graph such as variable use, function distance, and beacons, possibly with similarity learning to capture the code structure.

CAVA-CV utilizes model-tracing [9] to track a user's interactions in Ghidra and uses the activation equation from ACT-R [3] to generate levels of intensity for the highlighting module. The input to the model is a label representing a piece of code (e.g., a line, block, function, or other structured piece of information), henceforth referred to as a "line", and a timestamp. The activation A_i of a line is computed as follows:

$$A_i = ln \sum_{j=1}^{n} t_j^{-d} + MP * \sum_{k} Sim(v_k, c_k)$$

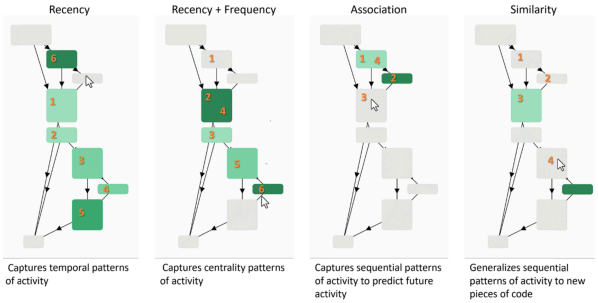

Figure 2. Examples of various forms of the cognitive visualizer (CAVA-CV) to inform highlighting. Numbers indicate the click sequence in each example.

The first term accounts for recency and frequency, where t is the time since the j^{th} visit to line i and d is a constant decay rate. The second term accounts for similarity between lines, where $Sim(v_k, c_k)$ is the dissimilarity between the current line and line i for feature k (usually linearly scaled between 0.0 and -1.0, from perfect match to very different). MP is a mismatch penalty parameter that scales the similarity value, and is set at default to 1.

The activation would be determined for each other line in comparison to the current line for the purpose of highlighting other relevant lines to the current one. As shown in Figure 2, multiple CAVA-CV versions have been developed to scale up in complexity and the kinds of cognitive mechanisms represented in the computations. In the first example, starting from the left, the model relies only on recency from the activation equation. This model captures temporal patterns of activity so that the most recent clicks are highlighted darkest, and fading further into the past. The second example combines recency with frequency. This model captures centrality in patterns of activity, not only highlighting more recently clicked pieces of code darker, but also those lines that are visited more often and therefore more important to the current task. These first two examples provide a historical visualization of a user's past activity, but do not rely on the similarity term from the activation equation. This kind of information can help alleviate working memory constraints by providing a user a visual external aid of where they have been investigating, thus offloading internal mental resources.

In the third example, the model makes use of associative knowledge. The model keeps track of sequential activity patterns (e.g., the last three click locations) to predict future mouse clicks. The fourth example incorporates similarity from the activation equation. This model can generalize sequential patterns of activity to new pieces of code that are similar to previous experience. These similarities can be defined by the structure of code, code semantics, beacon similarity, or any other kinds of information that could be deemed useful in relating one piece of code to another.

The last two models can be used to provide assistance to a user and help guide them to where they should visit next. For example, the model can be trained using an expert's sequence of actions that could then be used to guide novices to where an expert would likely visit given the novice's prior sequences of actions. Figure 3 is a snapshot of CAVA where blue and red colors represent past activity and recommendations, respectively, and lightness is based on the recency and frequency. The right-most panel displays the estimated four cognitive states, each of which is determined by combining the following physiological responses:

- Stress: GSR, heart rate, and pupil dilation
- Cognitive load: Task load (from EEG), inverse of O2Hb from fNIRS, and the eye blink rate
- Fatigue: Eye gaze entropy, blink rate, EEG alpha band, inverse of heart rate, and engagement (from keyboard and mouse clicks)
- Insight: P300 (from EEG) [18], pupil dilation, and event prediction

5. Implementation & Evaluation

We developed plugins to monitor user activities at different panels on Ghidra V9.0. CAVA release note describes the plugins in detail [15].

5.1. System Integration

Four main modules are involved in the closed-loop CAVA system: CAVA-CM, CAVA-CV, neurocognitive analysis pipeline, and Ghidra. During run time, the CAVA-CM module listens for UDP messages from the CAVA-CV module (e.g., mouse clicks), from the neurocognitive analysis pipeline (e.g., neurocognitive state data), and Ghidra events. Upon receiving a user activity, CAVA-CM processes it and returns a result to drive user-centered interventions. In return, CAVA-CV generates responses to highlight on Ghidra. The cognitive models are implemented in ACT-UP [19] that has been designed for scalability and computational efficiency.

5.2. Experiment with Reverse Engineers (REs)

Institutional Review Board (IRB) approved our research protocol that involves the following tasks and procedure.

RE tasks. Our experiment consists of 20 basic tasks and 6 advanced tasks. Basic tasks represent those that are continuously performed throughout reverse engineering tasks such as opening a specific program for analysis

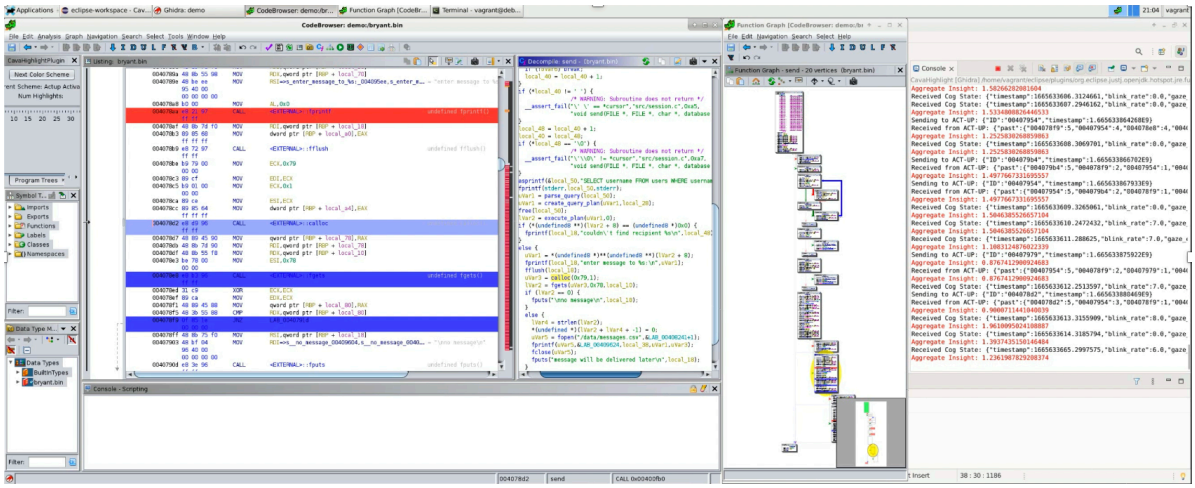

Figure 3. CAVA-CV in Ghidra. In the listing view, red represents recommendations as if an expert were guiding the current analyst, and blue represents the analyst's past activities based on recency and frequency. Color and lightness recommendations are driven by the cognitive states analyzed in real time by CAVA-CM (far-right plug-in).

in Ghidra, navigating to different locations within the target program, searching for ASCII strings, using cross-references to navigate the program, navigating into a called function, following a path in the program's control flow, analyzing and navigating cross references to a particular function, analyzing a program and indicating locations where a particular variable is used, relabeling generically-named function and parameter names, and determining the entry point for a program.

An advanced task consists of a Point of Interest (POI) triage, followed by Point of Vulnerability (POV) analysis and annotation on a target program in Ghidra. All tasks are from the Bryant challenge program [10] and in-house experts vetted task difficulties. During a POI triage, an analyst needs to determine an estimated value in continuing analysis. This task is procedurally unconstrained, and provides a direct assessment of the time taken to perform the initial triage as well as a measure of whether the participant makes a correct assessment. Our expectation is that the user will use some of the previous subtask techniques in performing their analysis and assessment, but the precise steps taken will require post-experiment interview with video playback to determine if finer-grained objective measures can be made and how prior experience transfers to a new POI/POV. When the analyst determines to continue to POV, (s)he performs careful analysis of a particular program segment and annotates instructions which are likely to be implicated in an exploitable program defect. This task provides a direct assessment of code comprehension performance and skill and post analysis can provide indications of correctness. Just as in the POI triage, we expect the user to make use of skills which were demonstrated when performing prior subtasks.

Procedure. We recruited participants with prior experience in reverse engineering; they were either government/military personnel without compensation, or employees who were provided a charge number for their time spent to participate in this experiment.

After signing an informed consent form, candidate participants responded to an online preliminary survey to assess their general knowledge in computer science and reverse engineering, as well as years of experience in vulnerability analysis and Ghidra usage. Among 12 who expressed interest, we invited 9 to proceed. An experiment consisted of placing neurophysiological sensors and calibrating them, following instructions in Ghidra to complete 26 tasks in total, and completing the post-experiment survey. Each experiment took approximately 3 hours.

5.3. Open-Loop Experiment

To support novice REs, an important assessment is distinguishing users according to their background knowledge, primarily based on neurophysiological and behavioral responses. We can then minimize unnecessary cognitive recommendations, which annoys users and prohibits new tools from adaption. To ensure that novices are distinguishable from experts using behavioral and neurophysiological responses, we conducted pilot open-loop experiments with 9 REs first. Table 3 summarizes their demographics. We assigned knowledge level based on their responses in the preliminary survey.

Time and response accuracy analyses. Figure 4 visualizes the average amount of time that participants spent on POI/POV tasks that they answered correctly and incorrectly, based on their prior knowledge level. In

Table 3. Participant demographics. RE and Ghidra columns represent the number of years of experience. We assigned RE knowledge (RE KNWL) based on their responses to a preliminary survey.

	Age	Deg.	RE	Ghidra	RE KNWL
P1	36–45	BS	< 3	< 3	Novice
P2	46–55	MS	< 3	< 3	Medium
P3	46–55	MS	< 3	< 3	Novice
P4	26–35	MS	< 3	< 3	Medium
P5	26–35	MS	< 3	None	Novice
P6	26–35	MS	3–6	6–10	Novice
P7	36–45	MS	< 3	3–6	Medium
P8	26–35	MS	< 3	< 3	Advanced
P9	56+	PhD	None	None	Novice

general, participants spent less time on tasks that they answered correctly compared to tasks that they answered incorrectly. For POI/POV tasks that they answered correctly, prior knowledge helped in reducing time spent on identifying vulnerabilities. We asked Likert-scale questions after each POV task regarding the confidence and use of strategies after each POV task (1:least – 5: most): (a) advanced indicated the highest confidence (5.3) compared to novices (3.4) and medium (2.5), and (b) advanced indicated the highest confidence (5) compared to novices (3.6) and medium (2.6). These results indicate that people with prior experience may have strategies to efficiently explore and exploit vulnerabilities, supporting findings by Votipka et al. [24].

Sample entropy analysis using eyetracking data. We analyzed eyetracking data that indicates cognitive states. Pupil dilation, for example, is closely tied to Noradrenaline activity in the Locus Coeruleus region of the brain and can be taken as a proxy for activation of the sympathetic nervous system [20]. To compute the entropy of the gaze locations on the screen during

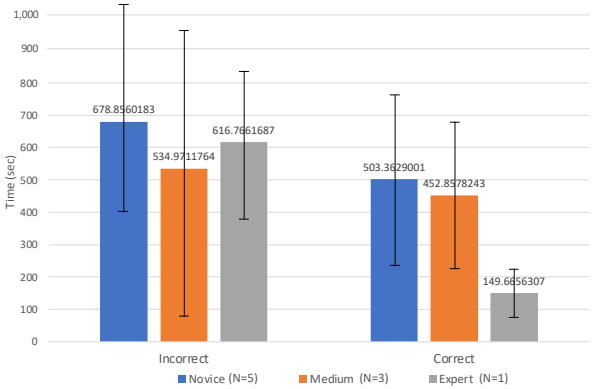

Figure 4. Average time that participants spent on POI/POV tasks. Participants spent more time on POV tasks that were incorrect, and participants' prior experience level is inversely related to the amount of time spent on tasks that they answered correctly.

the Ghidra tasks, we used Sample Entropy (SE), which is defined as the negative natural logarithm of the conditional probability that two sequences similar for n points remain similar at the next point [5].

SE can determine relative levels of expertise between different REs and relative difficulty of different problems, and we observed differences in SE values according to their knowledge levels on the POV responses (Figure 5). Novices spend ample time on specific regions that they are only familiar with, resulting in low entropy but high in the amount of time spent. On the other hand, experts know specific areas that might have vulnerabilities, hence, they scan through potential vulnerabilities very fast, resulting in high entropy in a short period of time.

Cognitive workload using fNIRS. The Functional Near-Infrared Spectroscopy (fNIRS) sensor was used to observe changes in cognitive load over the task sequence. This sensor is attached to the forehead and measures

Table 2. POV responses. BO, SQL, ML, and NP stand for buffer overflow, SQL injection, memory leak, and null pointer, respectively. The expertise column corresponds to pre-assigned expertise levels using the preliminary survey responses. Total represents the number of POVs that the corresponding participant answered correctly.

	POV_1	POV_2	POV_3	POV_4	POV_5	POV_6	Total	Knowledge
P1	TP: SQL	TP: BO	**FP**	**TP: SQL**	TP: BO	TP: BO	1	Novice
P2	**TP: BO**	TP: BO	TP: BO	FP	TP: BO	**FP**	2	Medium
P3	**TP: BO**	FP	TP: BO	**TP: SQL**	**FP**	TP: BO	3	Novice
P4	FP	FP	**FP**	**TP: SQL**	**FP**	**FP**	4	Medium
P5	FP	FP	**FP**	**TP: SQL**	TP: BO	Unsure	2	Novice
P6	FP	FP	**FP**	FP	TP: BO	Unsure	1	Novice
P7	FP	FP	TP: BO	**TP:SQL**	TP: ML	**FP**	2	Medium
P8	TP: SQL	FP	**FP**	**TP: SQL**	TP: BO	**FP**	3	Advanced
P9	TP	TP	**FP**	TP: BO	Unsure	**FP**	2	Novice
Answer	**TP: BO**	**TP: NP**	**FP**	**TP: SQL**	**FP**	**FP**		
Difficulty	D	M	E	D	D	M		

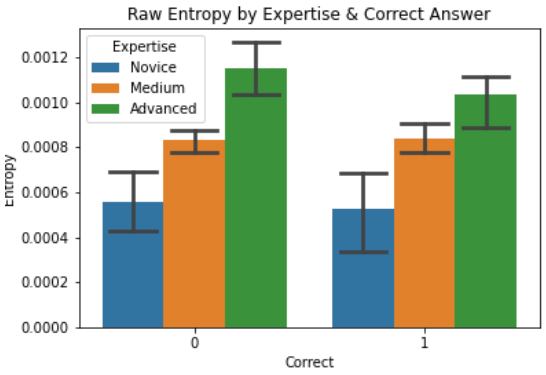

Figure 5. Sample entropy (SE) vs. knowledge. Eyetracking patterns show higher entropy on REs with reverse engineering knowledge compared to novices.

changes in concentrations of oxygenated hemoglobin (O2Hb) and deoxygenated hemoglobin (Hhb). An increase in O2Hb and decrease in Hhb is associated with task complexity and cognitive workload [7].

We applied z-score to the recorded O2Hb and Hhb values, and averaged sensor values for each task. Although each participant appears to have a different sensor response, a clear flip in polarity for one or both of the O2Hb and Hhb values was observed when the POI/POV tasks began. As participants switched from well-defined introductory tasks to more free-form POI/POV tasks, their workload could either increase if they find POV discovery challenging, or decrease if they found the interface familiarity tasks more challenging. Workload could also decrease if they found the POI/POV tasks too challenging and gave up.

In particular, P8 (advanced) displayed the expected pattern of increased O2Hb and decreased Hhb at the onset of the POI/POV tasks as shown in Figure 6, except for POI_5 and POI_6 when the fNIRS sensor died to result in constant O2Hb and Hhb values. P7's sensor values indicate that workload increased up until the POI/POVs, then dropped off. This output may be indicative of fatigue over the course of the POI/POVs, or could indicate that the participant was more familiar with POI/POV tasks than the Ghidra interface familiarity tasks, and therefore needed to exert less effort to understand them.

Task load and engagement using EEG. We conducted frequency band analysis on the collected EEG data at 500 Hz sampling rate to estimate task load and engagement. Signals were notch-filtered at 60 Hz to remove powerline artifacts and band pass-filtered between 1 and 128 Hz.

For each POI/POV task pair, we observed that the task load is significantly higher during POV, while engagement is significantly higher during the POI triage. For example, the engagement index for P6 is higher

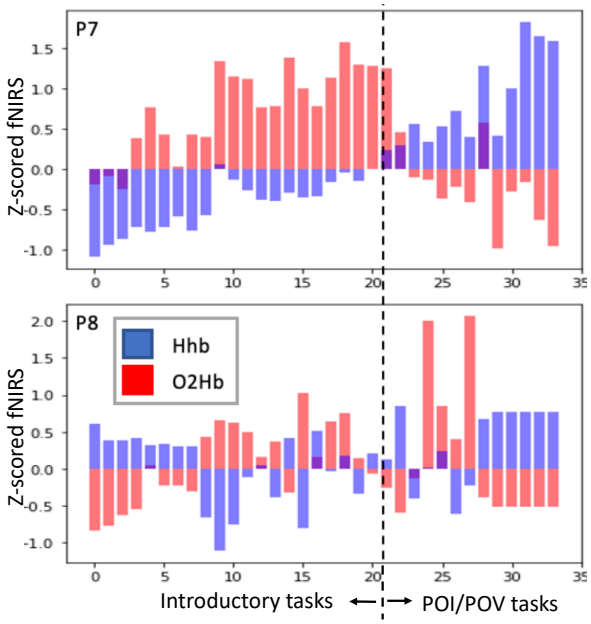

Figure 6. fNIRS responses and workload assessment. A flip in polarity for one or both of the O2Hb and Hhb values is observed when the POI/POV tasks with higher cognitive demand begin.

during triage, while task load is higher during analysis as shown in Figure 7. This pattern suggests that participants are more alert and attentive during triage, while they are searching for a vulnerability. Differently, while analyzing and describing a vulnerability, a higher task load indicates that participants may be storing more in working memory as they remember and describe the result of triage, explaining whether or not a vulnerability exists in the code. More insight can be gained from measures of engagement and task when one understands the details of the task. During a difficult task (e.g., POI_5 includes a large amount of complicated code but no vulnerability), many participants have significantly higher engagement during triage, while they search the complex code base for a vulnerability that does not exist. In-depth analyses can be found in another paper [16].

Various analyses results from physiological sensor responses suggest the possibility of inferring when an RE faces challenges in vulnerability discoveries, supporting the potential benefits of a closed-loop cognitive recommender system.

5.4. Pilot Closed-Loop Experiment

This pilot experiment involves CAVA-CV that provides real-time visual recommendations according to the estimated cognitive states of the analyst wearing neurophysiological sensors. The experiment procedure remained the same, and the analyst interacted with

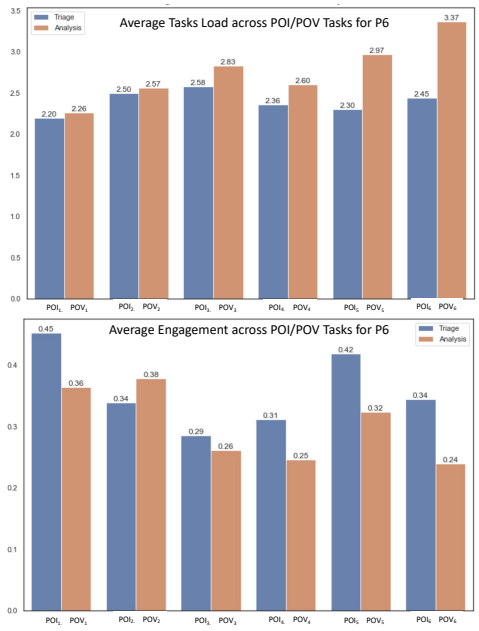

Figure 7. Task load (TL) and engagement index (EI) for POI/POV tasks. EI is higher during POI triages, while TL is higher during POV analyses.

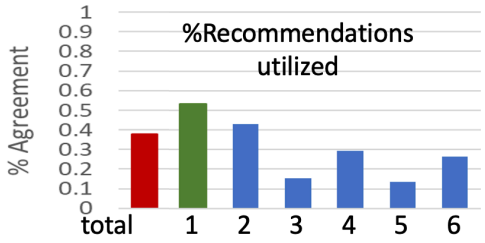

Figure 8. Utilization of CAVA recommendations. Our participant utilized 38% of CAVA recommendations.

the instrumented Ghidra as shown in Figure 3. Due to challenges in recruiting REs, we pilot-tested the closed-loop CAVA system with 1 participant. We compared the participant's behavior with one who has similar RE experience in the open-loop experiment.

Preliminary results. We observed high agreement between user actions and recommendations from CAVA-CM (38%) on 6 POI/POV tasks, reflecting possibility of trust and usability of cognitive visualization (Figure 8). POI_1, in particular, highlights effectiveness of expertise-based recommendations with 55% agreement.

We compared the mouse entropy when Ghidra displays recommendations to the baseline Ghidra interface. As shown in Figure 9, cognitive visualization directs the user's attention appropriately, indicating reduction in user confusion. We also observed significantly reduced time spent on POI/POV tasks using the cognitive recommendations, as shown in Figure 10.

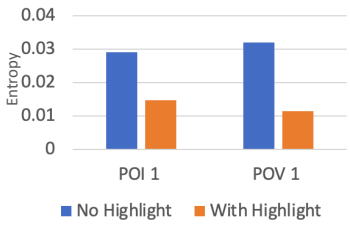

Figure 9. Mouse entropy for a POI & POV task. CAVA-CV directs the user's attention appropriately, reducing user confusion.

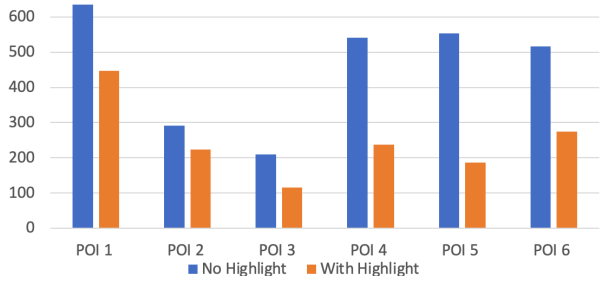

Figure 10. Time spent in POI tasks using CAVA-CV. CAVA-CV has potential to significantly reduce time analysts spend on POI tasks.

At the end of each task, we asked the participant to rank the usefulness of the recommendations using a Likert scale (1: not useful at all – 7: very useful). As shown in Figure 11, the participant indicated that the recommendations were generally useful, except for POI/POV_5 where he faced technical issues.

In terms of computational and communication efficiency, our closed-loop performance was under 250ms, which includes time to read multimodal sensor inputs, estimate cognitive states, and visualize recommendations. The participant did not notice any delay in receiving the visual recommendations.

6. Limitations

One major challenge was recruiting participants with prior experience in reverse engineering. Although we actively recruited participants from government, military, universities, and industries, we were only successful

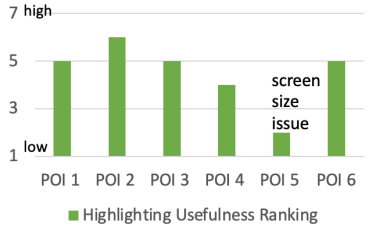

Figure 11. Usefulness ranking that indicates user acceptance in visualizing recommendations.

in recruiting 9 REs for the open-loop and 1 for the closed-loop experiment. Remote experiments can be offered at the cost of reduced accuracy. For example, we can ship some sensors (e.g., eyetracker, heart rate/GSR monitor) that participants can easily set up and calibrate. Without neurophysiological inputs, however, we will lose the accuracy of distinguishing cognitive states with similar physiological responses (e.g., stress vs. excitement) and may result in unnecessary visual recommendations. Identifying relationships between cognitive and physiological states and minimal sensor requirement are active research areas.

In the cognitive modeling approach, one current limitation includes lack of access to all available contextual information, and some of the outcomes/tool actions. For example, our current instrumentation cannot determine that a pop-up window (e.g., the relabel window) was activated given an input action or whether the user interacted with the window. Therefore, these contexts and tool actions will need to be inferred based on the input actions given a particular goal. Validating the inference process against video data of user interactions may increase the confidence in the cognitive model.

Since the GDM model in CAVA-CM learns experts' behavior and processes of tackling reverse engineering tasks, CAVA is generalizable to guide novices on tasks that the system has not been exposed to. Additional details of CAVA-CM models are in another paper [9], and we leave it as our future work to evaluate how CAVA adapts to novel tasks.

7. Related Work

We review related work that focus on understanding mental models, processes, and guidelines of REs (i.e., systems that automatically identify vulnerabilities or dynamically rewrite/obfuscate code is outside the scope). To understand REs' mental models and processes, Votipka et al. compared the descriptions of the RE processes between experts and novices [25], and found that both groups followed roughly the same steps. Fang et al. surveyed REs to identify the types of automation they use, and found that REs preferred dynamic over static analyses [11, 13]. Another interesting finding is that REs deal with ambiguity by discussing with others and relying on visualization techniques (i.e., mapping system semantics on a whiteboard).

Since reverse engineering is a complex task, REs tend to rely on the community for guidance and knowledge to achieve their goals. Votipka et al. investigated 1,590 discussions among 688 REs over Tweeter, Reddit, and StackExchange [23]. According to their investigation, REs are most interested in features for customizing Ghidra. They also observed limited evidence of collective sensemaking on the forums, with few REs participating in multiple discussions threads and most acting as either knowledge producers or consumers. They also found that the forums operated similarly, but Twitter was most often used to announce information (e.g., tutorial links, tool overviews, vulnerabilities in Ghidra) and REs used StackExchange mostly to get support for specific problems. Reddit acted as a middle option.

These findings confirm that REs can benefit from CAVA that provides experts' recommendations as real-time visual highlights. While closed-loop recommender systems have been proposed, prior work oftentimes lack realization [17, 4, 8, 22] or results are oftentimes based on simulation results [21]. Our work includes developing Ghidra plugins and running open-loop and closed-loop experiments with REs.

8. Conclusions

This paper is a first attempt to develop a closed-loop recommendation system that visualizes expert recommendations according to the RE's current cognitive states. When a novice RE starts experiencing fatigue, stress, and increase in cognitive load, CAVA guides the novice RE to where an expert would likely visit given the novice's prior sequences of actions. We developed plugins for Ghidra V9.0 to collect behavioral responses, and to incorporate ACT-UP cognitive model and cognitive visualizers. In particular, CAVA aims at minimizing computational and communication delays to analyze cognitive states and provide appropriate recommendations in real time. According to our pilot experiments with REs, our approach is promising in helping novices learn vulnerability analysis procedures, enhancing user trust, usability, and acceptance.

Acknowledgments

This work was supported by DARPA through AFRL Contract HR0011-20-C-0141. All statements of fact, opinion or conclusions contained herein are those of the authors and should not be construed as representing the official views or policies of DARPA, AFRL, or the U.S. Government. Distribution Statement "A" (Approved for Public Release, Distribution Unlimited).

References

[1] B. Abrath, B. Coppens, J. V. D. Broeck, B. Wyseur, A. Cabutto, P. Falcarin, and B. D. Sutter. Code renewability for native software protection. *ACM Transactions on Privacy and Security*, 23(4), 2020.

[2] J. R. Anderson, D. Bothell, M. D. Byrne, S. Doublass,

C. Lebiere, and Y. Qin. An Integrated Theory of Mind. *Psychological Review*, 111(4):1036–1060, 2004.

[3] J. R. Anderson and C. Lebiere. *The atomic components of thought*. Psychology Press, 1998.

[4] J. H. Barrow, C. L. Baldwin, D. M. Roberts, B. A. Taylor, C. Sibley, J. T. Coyne, A. Mandulak, G. Buzzell, and N. Penaranda. Using physiological measures to improve training for uav operators. In *16th International Symposium on Aviation Psychology*, 2011.

[5] G. Borghini, G. Vecchiato, J. Toppi, L. Astolfi, A. Maglione, R. Isabella, C. Caltagirone, W. Kong, D. Wei, and Z. Zhou. Assessment of mental fatigue during car driving by using high resolution eeg activity and neurophysiologic indices. In *IEEE Engineering in Medicine and Biology Society*, page 6442–6445, 2012.

[6] R. A. Bridges, S. Oesch, M. D. I. Miki E. Verma, K. M. Huffer, B. Jewell, J. A. Nichols, B. Weber, J. M. Beaver, J. M. Smith, D. Scofield, C. Miles, T. Plummer, M. Daniell, and A. M. Tall. Beyond the hype: A real-world evaluation of the impact and cost of machine learning-based malware detection. arXiv:2012.09214, August 2022.

[7] M. Causse, Z. Chua, V. Peysakhovich, N. D. Campo, and N. Matton. Mental workload and neural efficiency quantified in the prefrontal cortex using fnirs. *Scientific Reports*, 7(5222), 2017.

[8] J. T. Coyne, C. Baldwin, A. Cole, C. Sibley, and D. M. Roberts. Applying real time physiological measures of cognitive load to improve training. In *International Conference on Foundations of Augmented Cognition*, 2009.

[9] E. A. Cranford, C. Lebiere, R. Bhattacharyya, S. J. Fugate, D. Morrison, A. Barbiux, E. Kim, M. Levy, B. Marsh, J. Rego, M. Sayer, A. Waagen, F. Maldonado, J. P. Johnson, J. DiVita, and J. M. Buch. Cognitive models of reverse engineering processes. Manuscript in preparation, 2023.

[10] Cromulence. Chess challenges. `https://github.com/cromulencellc/chess-aces/blob/main/phase-1-report.pdf`, 2021.

[11] M. Fang and M. Hafiz. Discovering buffer overflow vulnerabilities in the wild: An empirical study. In *8th International Symposium on Empirical Software Engineering and Measurement*, 2014.

[12] D. Gilbert, C. Mateu, and J. Planes. The rise of machine learning for detection and classification of malware: Research developments, trends and challenges. *Journal of Network and Computer Applications*, 153(C), March 2020.

[13] M. Hafiz and M. Fang. Game of detections: how are security vulnerabilities discovered in the wild? *Empirical Software Engineering*, 21(5):1920—-1959, 2016.

[14] C. Lebiere, P. Pirolli, R. Thomson, J. Paik, M. Rutledge-Taylor, J. Staszewski, and J. R. Anderson. A Functional Model of Sensemaking in a Neurocognitive Architecture. *Computational Intelligence and Neuroscience*, 2013(5), 2013.

[15] F. Maldonado, J. P. Johnson, J. M. Buch, and S. J. Fugate. Ghidra CAVA Release Note. `https://github.com/niwcpac/cava`, 2023.

[16] B. Marsh, J. Rego, M. Levy, M. Sayer, A. Waagen, A. Barbieux, E. A. Cranford, D. Morrison, F. Maldonado, J. P. Johnson, J. DiVita, J. M. Buch, E. Kim, C. Lebiere, S. J. Fugate, and R. Bhattacharyya. Decoding internal decision making during reverse engineering tasks. Under submission, 2023.

[17] J. E. McCarthy. *Technology Enhanced Learning: Best Practices*, chapter Military Applications of Adaptive Training Technology. IGI Global, 2008.

[18] J. Polich. Updating p300: An integrative theory of p3a and p3b. *Clinical Neurophysiology*, 118(10):2128–2148, 2007.

[19] D. Reitter and C. Lebiere. A cognitive model of spatial path planning. *Computational and Mathematical Organization Theory*, 16(3):220–245, 2010.

[20] R. F. Rojas, E. Debie, J. Fidock, M. Barlow, K. Kasmarik, S. Anavatti, M. Garratt, and H. Abbass. Electroencephalographic workload indicators during teleoperation of an unmanned aerial vehicle shepherding a swarm of unmanned ground vehicles in contested environments. *Frontiers in Neuroscience*, 14, 2020.

[21] W. S. Rossi, J. W. Polderman, and P. Frasca. The closed loop between opinion formation and personalised recommendations. *IEEE Transactions on Control of Network Systems*, (1–12), 2021.

[22] C. Sibley. Adaptive training in an unmanned aerial vehicle: Examination of several candidate realtime metrics. In *Applied Human Factors and Ergonomics*, 2010.

[23] D. Votipka, M. N. Punzalan, S. M. Rabin, Y. Tausczik, and M. L. Mazurek. An investigation of online reverse engineering community discussions in the context of ghidra. In *IEEE European Symposium on Security and Privacy*, 2021.

[24] D. Votipka, S. M. Rabin, K. Micinski, J. S. Foster, and M. M. Mazurek. An observational investigation of reverse engineers' processes. In *USENIX Security Symposium*, 2020.

[25] D. Votipka, R. Stevens, E. M. Redmiles, J. Hu, and M. L. Mazurek. Hackers vs. testers: A comparison of software vulnerability discovery processes. *Proceedings of the IEEE*, 2018.