Combining Machine Learning and Cognitive Models for Adaptive Phishing Training

Edward A. Cranford (cranford@cmu.edu)
Department of Psychology, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15221 USA

Shahin Jabbari (shahin@drexel.edu)
Department of Computer Science, Drexel University
3675 Market Street, Office 1151, Philadelphia, PA 19104 USA

Han-Ching Ou (hou@g.harvard.edu)
Department of Computer Science, Harvard University
29 Oxford Street, Cambridge, MA 02138 USA

Milind Tambe (milind_tambe@harvard.edu)
Center for Research in Computation and Society, Harvard University
150 Western Avenue, Room 2.107, Allston, MA 02134 USA

Cleotilde Gonzalez (coty@cmu.edu)
Department of Social and Decision Sciences, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15221 USA

Christian Lebiere (cl@cmu.edu)
Department of Psychology, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15221 USA

Abstract

Organizations typically use simulation campaigns to train employees to detect phishing emails but are non-personalized and fail to account for human experiential learning and adaptivity. We propose a method to improve the effectiveness of training by combining cognitive modeling with machine learning methods. We frame the problem as one of scheduling and use the restless multi-armed bandit (RMAB) framework to select which users to target for intervention at each trial, while using a cognitive model of phishing susceptibility to inform the parameters of the RMAB. We compare the effectiveness of the RMAB solution to two purely cognitive approaches in a series of simulation studies using the cognitive model as simulated participants. Both approaches show improvement compared to random selection and we highlight the pros and cons of each approach. We discuss the implications of these findings and future research that aims to combine the benefits of both methods for a more effective solution.

Keywords: cognitive models; model-tracing; restless multi-armed bandit; Instance-Based Learning; ACT-R; phishing

Introduction

Phishing remains one of the biggest threats to cybersecurity in an organization (APWG Phishing Report, 2021). Typical training of employees involves limited cybersecurity awareness tutorials and simulation campaigns (Yeoh et al., 2021). During simulation campaigns, phishing emails are sent to employees, usually selected at random, and if a user clicks on a link embedded in the email, then they are given immediate feedback and training about how to detect phishing emails. While the method is effective compared to no intervention, it may be ineffective if it targets more phish-aware users than naïve users who are more susceptible to phishing. We believe that simulation campaigns could be improved through personalization by strategically selecting who to target. However, to determine who to target for training, one needs a representation of the cognitive states of each individual in the organization (i.e., their propensity to fall victim to a phishing attack).

Recent advances in simulation campaigns attempt to personalize training to determine which users to select based on risk propensity (e.g., Cyber Guru, 2019), but these approaches do not account for human experiential learning and adaptivity through repeated interactions with the environment. Recent research in end-user susceptibility to phishing emails (Cranford et al.., 2021) implies that phishing classification decisions can be framed as decisions from experience in accordance with Instance-Based Learning Theory (IBLT; Gonzalez, Lerch, & Lebiere, 2003). In line with IBLT, phishing decisions are made by retrieving classifications from memory and generalizing across past experiences, or instances, that are similar to the current email. Decisions are thus influenced by memory effects such as recency, frequency, and similarity of past emails to the features of the current email, and contribute to learning and adaptivity (e.g., Hakim et al., 2020; Singh et al. 2019; 2020).

The present research is a first step toward developing a training methodology that uses cognitive principles to determine what users to select to receive training at each time step. The problem can be framed as a scheduling problem that aims to optimize the targeting of users in order to maximize the overall probability of adopting safe email behavior, without bombarding users with interventions. Our solution combines cognitive models of end-user susceptibility to phishing emails and machine learning methods to identify users most in need of training. We use the Restless Multi-Armed Bandit (RMAB) framework that models each user (arm) as a Markov Decision Process (MDP), using the cognitive model to define the MDP. RMABs have been used successfully in healthcare settings to strategically assign intervention to patients most in need (Biswas et al., 2021), and anti-phishing training presents an analogous situation.

We also present a purely cognitive approach that incorporates model-tracing techniques to trace user behavior and identify which users to select at each time step. The RMAB solution is compared to the cognitive solution in a set of simulation studies using cognitive models as simulated participants. The results show that both approaches are equally more effective than random selection but differ in selection preferences. We highlight the pros and cons of each approach and discuss plans for future research that aims to combine the strengths of the MAB and cognitive approaches.

Modeling a Phishing Training Task

The task was designed to replicate a real-world phishing training scenario that could still be implemented in a human laboratory experiment. Users are run simultaneously in batches and are presented either a phishing email or a ham email on each trial as determined by the selection algorithm. Ham emails are non-spam, non-phish, “good” emails, intended for the specific recipient with a legitimate purpose. After each trial, users are provided feedback only after incorrectly classifying a phishing email, which represents immediate phishing awareness training from an organization, while users do not typically receive feedback otherwise.

While human subjects’ experiments are greatly limited by the number of users that can be run simultaneously in a laboratory setting (e.g., 10 is a practical number), simulations are less restrictive. Therefore, in all reported analyses, we simulated 1000 users (near maximum possible for parallel simulations with 16GB RAM) for 100 trials of training (near maximum trials possible in a 1-hour laboratory experiment).

Defining Users

Among the vast individual differences and factors that influence phishing susceptibility, including demographics such as age, sex, and education (e.g., Sheng et al., 2010), and personality and social factors such as the Big 5 or the Dark Triad (Curtis et al., 2018; Yang et al., 2022), one of the most important factors is amount of email usage and knowledge and experience with phishing emails and network security (Lin et al., 2019; Sheng et al., 2010; Yang et al., 2022). In fact, these factors of overall *email usage* and *phishing and network security experience* align well with our own theory that defines user susceptibility to phishing as arising from decisions from experience as outlined by IBLT. Therefore, we designed a set of users that we could simulate in our IBL model based on individual differences in initialized instances. Each user in the model is initialized with a random number of emails (10-100 in increments of 10, uniformly distributed; Initialized Length), which represents individual differences in the amount of *email usage*, of which a random proportion are phishing emails (0.7-1.0, normally distributed within limits and rounded to the nearest 0.05, *M* = 0.85, *sd* = 0.05; Ham Proportion), which represents individual differences in the amount of *phishing and network security experience*. We used the same set of users in all simulations reported below.

Cognitive Model Description

Cranford et al. (2021) developed a generalizable IBL model of phishing susceptibility as arising from decisions from experience. The model accurately predicted classification decisions in two different tasks with different databases of phishing and ham emails: the Phishing Training Task (PTT; Singh et al., 2019) and the Phishing Email Susceptibility Test (PEST; Hakim et al., 2020). This model was used in the simulations reported below to generate predictions of human decision making against each selection algorithm and served as a basis for designing the Cognitive Selection algorithms.

The cognitive model was developed in ACT-R (Anderson & Lebiere, 1998) and makes classification decisions in accordance with the IBL process. On each trial, the model generates a classification decision by retrieving similar past instances based on the context features of the email. The features of the emails include the sender, subject, body, link text, and url. Decisions are thus based on the semantic similarity between email features. The semantic similarity values between features of two emails are computed using the University of Maryland Baltimore County’s semantic textual-similarity tool (Han et al., 2013), which uses a combination of latent semantic analysis (LSA) and WordNet. Retrieval of past instances is based on ACT-R’s blending mechanism (Lebiere, 1999; Gonzalez et al., 2003) which returns a consensus value (in this case, a classification of ham or phish) across all memories, rather than from a specific memory:

$$V = \underset{V_o}{\operatorname{argmin}} \sum_i P_i \times \left(1 - Sim(V_o, V_i)\right)^2 \qquad (1)$$

The value V is the one that minimizes the dissimilarity between the possible decisions and the actual decision in chunk i, weighted by the probability of retrieval P_i of the matching chunk *i* in memory.

$$P_i = \frac{e^{A_i/t}}{\sum_j e^{A_j/t}} \qquad (2)$$

P_i reflects the ratio of an instance’s activation A_i and temperature *t*, which defaults to $\sqrt{2} * s$, where s equals the variance parameter of noise. The activation A_i of an instance *i*, is determined by:

$$A_i = \ln \sum_{j=1}^{n} t_j^{-d} + MP * \sum_k Sim(v_k, c_k) + \varepsilon_i \qquad (3)$$

where the first term reflects the power law of practice and forgetting, where t_j is the time since the *j*th occurrence of chunk i and d is the decay rate (set to 0.5). The second term reflects the sum of similarities of each contextual feature k for the current item c and the corresponding element in memory chunk v, weighted by the mismatch penalty MP (set to 2.0). The final term represents noise, a random value from a normal distribution with mean of zero and variance *s* of 0.25, and introduces stochasticity in retrieval.

After making a classification, the instance is saved to memory and influences future decisions. However, if the email was a phishing email and it was incorrectly classified, the user is given feedback, and the decision is changed from ham to phishing to reflect the ground truth classification.

Multi-Armed Bandits Selection Algorithm

The MAB problem is a well-studied online machine learning setting. In the classic problem, also known as stochastic MAB (Cesa-Bianchi & Lugosi, 2006), in each round, the learner (here the security team of the company) selects an arm (here an employee of the company) for an intervention (here sending a phishing email) and receives feedback (here the proficiency of the participant against the phishing attack) which is typically referred to as the reward. This process continues for a fixed number of rounds (referred to as the time horizon) and the goal is to maximize the total reward observed by the learner.

The classic setting assumes the arms are static such that the distribution of rewards for each arm remains stationary regardless of past arm selections. This is not the case in our setting, as users react to training and potentially become less vulnerable to future phishing attacks. Various extensions to MAB have been proposed in the literature to model these reward distribution changes. The most general framework to model such scenario is what is known as the RMAB (Whittle, 1998) in which each arm is modeled as an MDP.

Since each arm represents an employee in our problem, the MDP can be used to model the progress of an employee throughout training. In general, an MDP is a quadruple consisting of (1) states (here the different degree of proficiency of the employee in detecting phishing attacks), (2) actions (here whether the training has been provided for the employee or not), (3) rewards or the value associated with being in each of the states (here whether or not the phishing attack can fool the employee in the employee's current state of proficiency) and the (4) transition probabilities which is a distribution over the possible next states given the current state and the chosen action (here how proficiency can change given the current level of proficiency and whether a training has been performed or not).

In our problem, we propose the following stylized MDP to model an employee. We assume there are two states, referred to as "good" and "bad" states. We further assume that there only two actions: a training intervention (action 1) and no intervention (action 2). The rewards for being in a good or bad state are assumed to be 1 and 0, respectively. The employee-dependent transition probabilities can be succinctly represented by 4 parameters: p_{gb}^1, p_{gb}^2, p_{bg}^1, and p_{bg}^2, where p_{xy}^i denote the probability of transfer from state *x* to state *y* when action *i* is taken.[1]

We used the cognitive model, described above, to generate the transition probabilities for each user cluster that were needed for the MDP. We simulated 1000 cognitive agents performing the task paired against a random selection algorithm. We defined a good state as a correct classification, and a bad state as an incorrect classification. Based on the model's sequence of decisions, probabilities were computed as the proportion of transitions from a good or bad state at time *t* to a good state at *t*+1 as opposed to a bad state at *t*+1, depending on the action (i.e., type of email sent) at time *t*.

While cognitive architectures and Markov Decision Processes (MDP) are quite different modeling approaches, they also share substantial similarities. Both embody the Markovian assumption of future behavior being probabilistically determined by the current state of the system and inputs from the environment. However, the current state for cognitive architectures consists of knowledge and skills held in memories, together with their activation, enabling both a more graded and combinatorial representation. Also, state transitions in cognitive architectures are largely determined by constrained mechanisms resulting from a theory of cognition, rather than needing to be trained from data. Therefore, unlike MDPs, cognitive architectures can make a priori predictions in the absence of data (Lebiere et al, 2003). Cognitive architectures can then be used to provide a high-fidelity model of human behavior on a limited set of available data, then run many times over new generalization conditions to provide large data sets for training MDPs (Sycara et al, 2015).

We highlight that in our formulation, while the states, the actions and the rewards are known, the transition probabilities for each of the employees are unknown and should be learned during the learning process. In general, RMAB problems are computationally hard and optimal solutions are only known for specific cases. We build on Whittle Index Q-Learning (WIQL), a recent algorithm proposed by Biswas et al. (2021), to design an algorithm which we call SuperArm-WIQL to solve our formulation of the RMAB problem. Intuitively, had we known everything about the MDPs in the RMAB problem, we could have used heuristic algorithms such as Whittle Index (Whittle, 1998) to decide which employee to target for intervention on any given round.[2] Without knowing the MDPs, one can use any off-the-shelf algorithm to simultaneously learn the parameters of the MDPs first before applying the Whittle Index heuristic. Biswas et al. (2021) use Q-Learning for this process and hence the name WIQL.

[1] Since there are two states and two actions, it seems like to fully represent the transition probabilities we require 8 parameters. However, observe that $p_{xy}^i + p_{xx}^i = 1$ for all states *xy* and action *i* as the transition will finally move to either of the two available states. Therefore, we can reduce the total parameter to only 4.

[2] We ignore the issue of indexability and conditions in which the Whittle Index heuristic is optimal.

The downside of such an approach is that learning the parameters of the MDP for each employee separately will result in a time and computational cost which is proportional to the number of employees. In practice, each round of sending phishing emails is costly and furthermore, the amount of available phishing emails is limited. Hence, naively applying WIQL will be too time-consuming, slow, and impractical. To deal with this problem, we first cluster the employees (or arms) into different groups (or super arms) and combine the learning experiences of all the users together. We call this algorithm SuperArm-WIQL. In the extreme, where there is only one arm per group, SuperArm-WIQL reduces to WIQL but with a small number of groups (compared to the total number of employees) and sufficiently similar arms in each group, SuperArm-WIQL will converge to a good policy much quicker.

We performed a K-means cluster analysis on the set of users described above to minimize the within-cluster sum of squares based on the Initialized Length and Ham Proportion attributes. A scree plot revealed four clusters were optimal ($SS_{bet}/SS_{tot} = 71.67\%$). Figure 1 shows the visualization of the four clusters, which we labeled according to their location in the landscape of Initialized Length and Ham Proportion: 1 = “high-high”, 2 = “low-low”, 3 = “low-high”, and 4 = “high-low”.

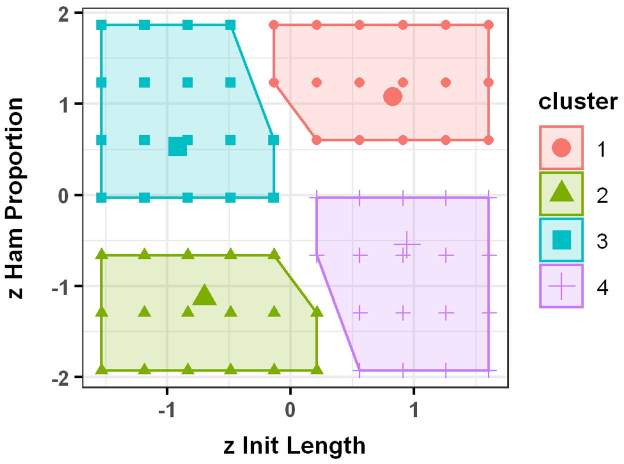

Figure 1: Cluster plot of simulated users

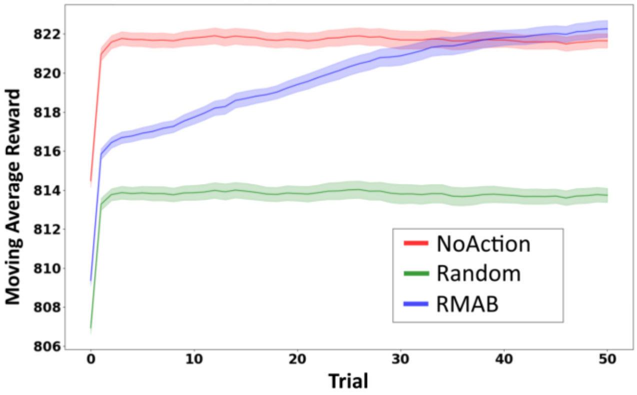

Figure 2: RMAB simulation results.

Simulation Results

The results of the RMAB simulation using the SuperArm-WIQL are presented in Figure 2, compared to Random and NoAction (no users selected for intervention) selection algorithms. Rewards are calculated as the sum of users in a good state (i.e., correctly classifying a given email) at each trial, and the plot shows the moving average reward with a window size of 50. To start the simulations, users are randomly assigned to states with 50% probability, and quickly transition toward good states. The NoAction and Random algorithms show that performance quickly plateaus as users align with the average transition probabilities given the possible actions. The results of the NoAction algorithm are a bit misleading because it only measures user proficiency in classifying ham emails (which is already high) and does not account for proficiency with phishing emails. Most notably, the results show that by selecting users strategically, the RMAB (blue) outperforms the Random algorithm (green) in terms of the number of users in good states, and continues to improve across trials, eventually outperforming the NoAction algorithm (red).

Cognitive Selection Algorithms

We designed two versions of the cognitive selection algorithm. The cognitive selection algorithms use cognitive principles to select which users to send phishing emails to on each trial, given a budget of 20% on each trial. Both methods use a technique called model tracing to track a user’s history of decision making (e.g., Anderson et al., 1995). For each trial, the algorithms store information about what email was presented to each user and what their decision was. This history is then used in the blending equation described above to compute probabilities of classifying an email as ham (V_{ham}) or phishing (V_{phish}), without adding any noise ε_i.

The first method, Cog-Low, simply computes the overall probability of classifying an email as ham or phishing at time t, without using the partial matching term. Therefore, the probabilities only reflect the influence of recency and frequency of all past instances. The participants with the lowest probability of classifying an email as phishing are selected for intervention (i.e., are sent a phishing email), with the hypothesis that their future probability of classifying phishing emails correctly will improve. The algorithm thus seeks to always improve the worst users on each trial.

The second method, Cog-EV, uses a more complex calculation that weighs the anticipated future benefits of sending a phishing email, in terms of correctly classifying phishing emails, against the anticipated future costs, in terms of incorrectly classifying ham emails, to determine which users will most benefit from a phishing training intervention.

As another improvement over Cog-Low, Cog-EV includes the partial matching term to determine the probabilities of correctly classifying an email of category k (ham or phish). Similarities are computed by averaging across the similarity of instance i to all other instances of the same category k. After computing the initial probabilities, another phishing instance is added to the user’s history to compute the future probabilities of correctly classifying a ham or phishing email given a phishing intervention. The expected value for sending a phishing email ($EV_{intervention}$) is reflected by the equation:

$$EV_{intervention} = \begin{array}{l}(V^{t+1}_{phish|intervention} - V^{t}_{phish}) - \\ (V^{t}_{ham} - V^{t+1}_{ham|intervention})\end{array} \qquad (4)$$

where V_{phish} and V_{ham} are the probabilities of correctly classifying a phishing or ham email, respectively, and are derived via blending.

Cognitive Simulations

We used instances of the cognitive model as simulated users to predict the effectiveness of the selection algorithms against humans. All simulations were seeded with the same initial random state and started with the same set of initialized users to ensure consistent replication. We used ACT-R's built-in mechanism for running multiple models in parallel. The selection algorithm determined which user to send phishing emails to on each trial. To minimize repeated presentation of emails per user, we used the 186 phishing emails from the PTT but combined the ham emails from both the PTT and the PEST, for a total of 177 ham emails. We compared the RMAB, Cog-Low, and Cog-EV algorithms to two baseline algorithms, NoAction and Random (random selection from a uniform distribution), resulting in 5 total conditions.

Results

The moving average accuracy across trials, with a window size of 50, is presented in Figure 3. The NoAction condition represents the high baseline accuracy in classifying ham emails correctly given no phishing training intervention. Between all other conditions, the RMAB and Cog-EV conditions perform best in terms of overall accuracy, but there is an interaction between phishing and ham accuracy such that phishing accuracy increases at the expense of ham accuracy. This reflects the tradeoff in signal detection due to frequency and recency effects.

Phishing accuracy improves the least in the RMAB condition, while the Random, Cog-Low, and Cog-EV conditions display similar improvements. However, the RMAB and Cog-EV conditions display the least decline in ham accuracy, while there is a greater decrease in Random. and more so in Cog-Low. These results are however difficult to interpret because they do not reflect differences in user selection preferences. It is possible that some algorithms are sending users the type of email that they are most likely to get correct, thus artificially inflating the overall accuracy. Therefore, we examined which users are being sent phishing emails as well as unbiased signal detection measures.Figure 4 shows a scatterplot of the mean accuracy for phishing and ham emails for each user, colored according to the proportion of phishing emails received, which is normalized within each selection condition (*z*-score). The results reveal distinct selection profiles. Accounting for the distribution of phishing emails across clusters, depicted in Figure 5 (*z*-scored phishing proportions), the Random condition displays no selection preferences and user accuracy trends with their phishing proportion. The RMAB selects users with high email experience and most phishing emails (high-low), which incidentally are already good at classifying phishing emails, while users that are poor at classifying phishing but good with ham emails receive more ham emails (top left tail of scatterplot). The Cog-Low mostly selects users with high experience and fewest phishing (high-high) which hypothetically need the most intervention, while sending the fewest phishing emails to the group that needs least intervention (low-low). The Cog-EV mostly send phishing emails to the users with low email usage (low-low and low-high), which are ones in which a training intervention will be most impactful, while sending the fewest phishing emails to the high-low group. However, there are a number of users that receive many phishing emails and thus their ham accuracy suffers (bottom right tail of scatterplot). If false alarms are not costly for a user or organization (i.e., by not responding important emails or causing excessive verification work for the security team) then this may be an acceptable solution.

Finally, to get a sense of the overall improvement of users from the start of the training task ("Initial" state) to the end of the training task ("Final" state). We examined change (Δ) in *d*-prime scores from the first 20 trials of the task to the last 20 trials of the task. We used a loglinear adjustment to account for missing cells when computing the hit rates and false-alarm rates (Stanislaw & Todorov, 1999). The results in Figure 6 show that Random selection improves sensitivity for

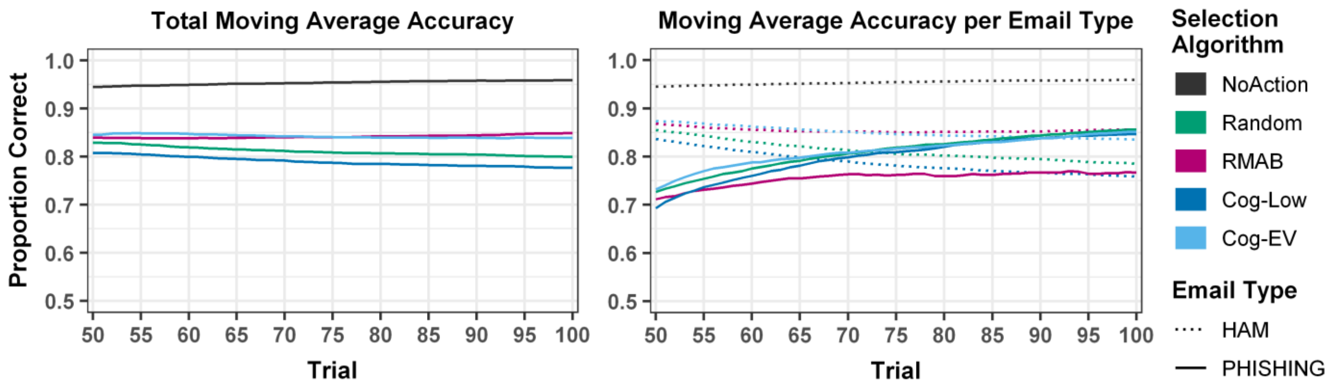

Figure 3: Moving average accuracy across trials for each selection condition. Total (left) and by Email Type (right). ws = 50.

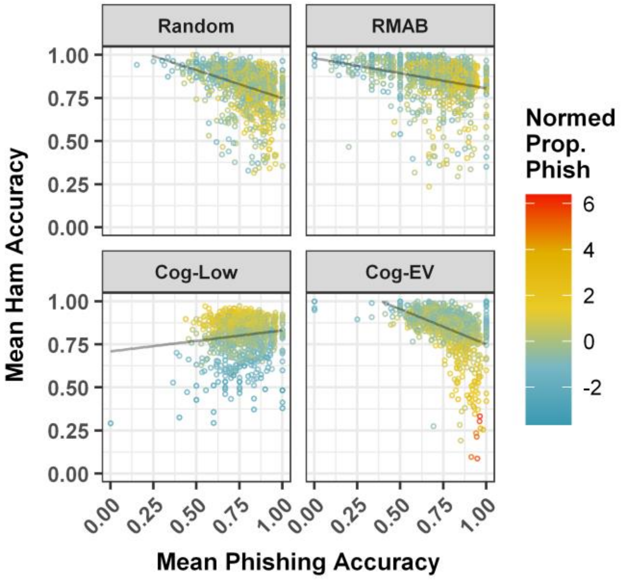

Figure 4: Scatterplot of individual ham and phishing accuracy colored by the normalized proportion (*z*-score) of phishing emails received within each selection condition.

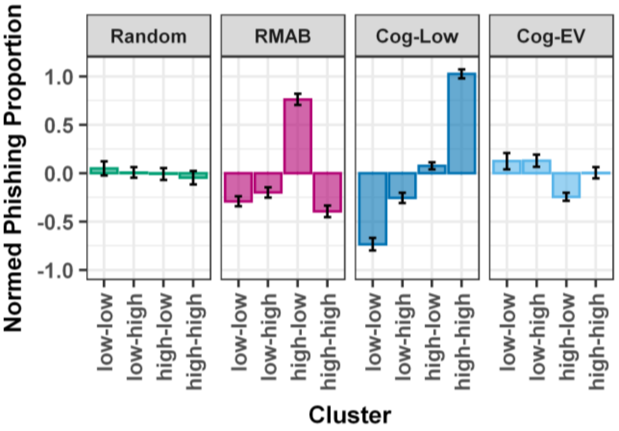

Figure 5: Normalized mean proportion (*z*-score) of phishing emails sent to each cluster within each selection condition.

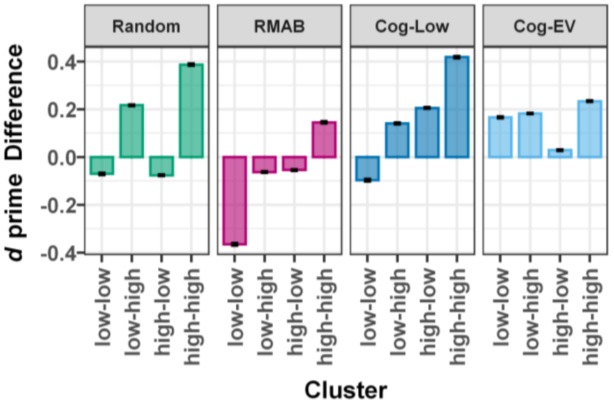

Figure 6: Delta *d*-prime scores from Initial state to Final state for each cluster within each selection condition.

users with lowest ham experience (high-low and high-high). The RMAB only improves the high-high even though they received the fewest phishing emails, but performance declines significantly for the low-low group. The Cog-Low improves performance more as the number of phishing emails presented increases. And lastly, Cog-EV is the only condition that improves sensitivity across all clusters.

Conclusion

Our simulations demonstrate the benefits of personalized anti-phishing training for organizations. The cognitive model proved useful in estimating transition probabilities for the MDP, and the RMAB was effective at improving performance. However, selection preference analyses revealed potential shortcomings of each of the methods. For one, the reward function for the RMAB should be redesigned so that it learns to send phishing emails to those most in need of intervention instead of those doing well. Current research is exploring methods such as defining states in terms of only phishing accuracy, but this would only lead to improvements in phishing classification. Another method could be to define rewards in terms of the users that misclassify emails (i.e., rewarded for intervening on those users that needed it).

Overall, the Cog-EV algorithm proved most successful at increasing phishing detection while minimizing false alarms. Future research will aim at validating these simulation results in human laboratory experiments. One limitation of the current simulations is that users were only given phishing emails as training interventions. However, it may be more realistic for users to receive phishing emails with some small probability in non-intervention events. We will consider this design change and its implications for selection algorithms.

The cognitive solutions have lower computational overhead and thus an advantage of selecting users at the individual level, while the RMAB is limited to generalizing at the group level. It is likely that the RMAB would perform better as the number of clusters approaches the number of users. Therefore, future research is aimed at finding the optimal tradeoff between the number of clusters and computational costs. Future research is also aimed at implementing a method that takes advantage of the benefits of both RMAB and cognitive models. For example, the cognitive model could be used to provide updated transition probabilities or additional learning rate parameters that can be used by the RMAB. Such an approach could both alleviate computational costs for the RMAB while providing more accurate predictions of individual behavior than Q-learning.

Finally, in other future research we plan to investigate not only whom to target but also which specific email to send and how to tailor emails to an individual. Such an approach could leverage information about what email features an individual is most susceptible to (e.g., Singh et al., 2020) or the type of attack for which they are most likely to fall prey (e.g., email characteristics or social engineering strategy used, or topic relevance; De Kimpe et al., 2018; Lin et al., 2019; Parsons et al., 2019). In this sense, IBL cognitive models are perfectly suited for every aspect of personalized anti-phishing training.

Acknowledgments

We thank Susobhan Gosh for their contribution of the code which we adapted to run our RMAB solution.

This research was sponsored by the Army Research Office and accomplished under MURI Grant Number W911NF-17-1-0370.

References

Anderson, J. R., Corbett, A. T., Koedinger, K., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of Learning Sciences, 4*, 167–207.

Anderson, J. R., & Lebiere, C. (1998). *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

Anti-Phishing Working Group (2021). *Phishing Activity Trends Report: 4th Quarter 2021*. Retrieved from https://apwg.org/trendsreports/. APWG.

Biswas, A., Aggarwal, G., Varakantham, P., & Tambe M. (2021). Learn to intervene: An adaptive learning policy for restless bandits in application to preventive healthcare. *Proceedings of the 30th IJCAI*, (pp. 4039–4046).

Cesa-Bianchi, N., & Lugosi, G. (2006). Prediction, learning, and games. *Cambridge University Press*.

Cranford, E. A., Singh, K., Aggarwal, P., Lebiere, C., & Gonzalez, C. (2021). Modeling phishing susceptibility as decisions from experience. *Proceedings of the 19th Annual Meeting of the ICCM* (pp. 44–49). Virtual.

Curtis, S. R., Rajivan, P., Jones, D. N., & Gonzalez, C. (2018). Phishing attempts among the dark triad: Patterns of attack and vulnerability *Computers in Human Behavior, 87*, 174-182.

Cyber Guru Phishing (2019). *Experiential learning to reduce phishing risk.* Retrieved from https://www.cyberguru.eu/cyber-guru-phishing/

De Kimpe, L., Walrave, M., Hardyns, W., Pauwels, L., & Ponnet, K. (2018). You've got mail! Explaining individual differences in becoming a phishing target. *Telematics and Informatics, 35*(5), 1277–1287.

Gonzalez, C., Lerch, J. F., & Lebiere, C. (2003). Instance based learning in dynamic decision making. *Cognitive Science, 27*(4), 591-635.

Han, L., Kashyap, A. L., Finin, T., Mayfield, J., & Weese, J. (2013). UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. In *Proceedings of the 2nd JCLCS* (pp. 44-52). Atlanta, GA.

Hakim, Z.M., Ebner, N.C., Oliveira, D.S. *et al.* (2020). The Phishing Email Suspicion Test (PEST) a lab-based task for evaluating the cognitive mechanisms of phishing detection. *Behavioral Research Methods*.

Lebiere, C. (1999). A blending process for aggregate retrievals. *Proceedings of the 6th ACT-R Workshop*. George Mason University, Fairfax, Va.

Lebiere, C., Gray, R., Salvucci, D. & West R. (2003) Choice and learning under uncertainty: A case study in baseball batting. *Proceedings of the 25th Annual Meeting of the CSS*. (pp. 704-709). Mahwah, NJ: Erlbaum.

Lin, T., Capecci, D. E., Ellis, D. M., Rocha, H. A., Dommaraju, S., Oliveira, D. S., & Ebner, N. C. (2019). Susceptibility to spear-phishing emails: Effects of internet user demographics and email content. *ACM Transactions on Computer-Human Interaction, 26*(5), 1–28.

Parsons, K., Butavicius, M., Delfabbro, P., & Lillie, M. (2019). Predicting susceptibility to social influence in phishing emails. *International Journal of Human-Computer Studies*, *128*, 17–26.

Sheng, S., Holbrook, M., Kumaraguru, P., Cranor, L. F., & Downs, J. (2010). Who falls for phish? a demographic analysis of phishing susceptibility and effectiveness of interventions. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (pp. 373–382).

Singh, K., Aggarwal, P., Rajivan, P., & Gonzalez C. (2019). Training to detect phishing emails: Effect of the frequency of experienced phishing emails. *Proceeding of the 63rd International Annual Meeting of the HFES.* Seattle, WA.

Singh, K., Aggarwal, P., Rajivan, P., & Gonzalez, C. (2020). What makes phishing emails hard for humans to detect? In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 64*(1). Chicago, IL.

Stanislaw, H., & Todorov, N. (1999). Calculation of signal detection theory measures. *Behavior Research Methods, Instruments & Computers*, 31(1), 137–149.

Sycara, K., Lebiere, C., Pei, Y., Morrison, D., Tang, Y., & Lewis, M. (2015). Abstraction of analytical models from cognitive models of human control of robotic swarms. *Proceedings of the 13th ICCM.* Groningen, NL.

Whittle, P. (1998). Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability*, *25,* 287–298.

Yang, R., Zheng, K., Wu, B., Li, D., Wang, Z., & Wang, X. (2022). Predicting user susceptibility to phishing based on multidimensional features. *Computational Intelligence and Neuroscience, 2022(*7058972), 1–11.

Yeoh, W., Huang, H., Lee, W-S, Jafari, F. A., & Mansson, R. (2021). Simulated phishing attack and embedded training campaign. *Journal of Computer Information Systems, 1–20.*