Using Machine Learning to Increase NPC Fidelity with Dynamic Preferences Used in Forward-Looking Decisions

Dustin D. Updyke
Thomas G. Podnar
Geoffrey B. Dobson

18 FEB 2021

TECHNICAL REPORT

CERT Division

[Distribution Statement A] Approved for public release and unlimited distribution.

Copyright 2021 Carnegie Mellon University.
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.
The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.
This report was prepared for the SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.
[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.
Internal use:* Permission to reproduce this material and to prepare derivative works from this material for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.
External use:* This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other external and/or commercial use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.
* These restrictions do not apply to U.S. government entities.
CERT® is registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.
DM21-0167

Abstract

//TODO

[Distribution Statement A] Approved for public release and unlimited distribution.

Introduction

This report summarizes our experience in modeling decision-making preferences within a distributed system of simulated users for cybersecurity training and exercise[1]. It also provides detail of our explorations of machine learning solutions and how we ultimately enabled increasingly life-like computer activity to autonomous agents over time. In addition, the particular problem we discuss is well-suited to others looking to build an introductory implementation of machine learning, in that our solution requires neither significant investment of money, resources, or data. We hope that operations teams will consider the problems they encounter in the same spirit as our approach, whereby the solution to the problem continues to improve itself in an automated fashion over time.

Since 2010, CERT Cyber Workplace Development (CWD) researchers have continually defined the crucial need for realism within elite cyber team training and exercise events. Our approach to the construction and execution of these events builds on the knowledge, skill, and experience of both participating individuals and teams in a measurable and repeatable fashion [Hammerstein 2010]. From the experience of executing hundreds of high-fidelity events with this approach, we went on to publish a design framework for cyber warfare exercises named Realistic - Environment, Adversary, Communications, Tactics, and Roles (R-EACTR) [Dobson 2017], providing further specifics on realism and focusing on its impact within training and exercise events.

In this spirit, we then introduced the GHOSTS framework [Updyke 2018], establishing a methodology for building behaviorally accurate, autonomous non-player characters (NPCs) within a high-fidelity cybersecurity training module or exercise. This effort grew out of our need to replicate the behavior of many different types of users on an enterprise network — both blue and red team users, and their specific activity — from the applications typical network users would employ, down to the network traffic they would generate. We went on to release a corresponding open-source software project — GHOSTS[2], which continues to grow and expand due to a community of users and contributors. This software remains under active development from within our CWD team, and from the support of the open-source training and exercise community.

We maintain that there exists a high potential value for the future delivery of ever-increasing fidelity with exercise scenarios. In this, there is likely a requirement for the complex coordination of distributed agents and their activities. It is also likely that machine learning could hold the key to ultimately building a thinking teammate or adversary.

---

[1] Much of what is discussed here is open source and part of GHOSTS SPECTRE: https://github.com/cmu-sei/GHOSTS-SPECTRE

[2] https://github.com/cmu-sei/GHOSTS

[Distribution Statement A] Approved for public release and unlimited distribution.

Note that our use of the word NPC hopes to span many other terms in the synthetic user domain. “User sim”, “network traffic”, etc. all point to the need for high-fidelity, lifelike activity on a computer network to properly simulate what users might be doing on that network and to create accurate network traffic that is the substrate of cybersecurity, the tooling around it, and the focus of much of training and experience. But while others do so from a perspective of network traffic first, we have always maintained that the best cyber teams triangulate their findings via said traffic, but also logs, sensor data, and a growing host-based toolchain. Thus, we have continued to focus on overall activity realism, and have used a litmus of comparing any NPC's activity to that of what we "would see by looking over someone's shoulder as they performed the same activity".

While the need for increasing fidelity of user simulation is not new, it remains a priority interest within particular areas of the military. In 2019, United States Cyber Command released challenge guidance for 39 critical areas within cybersecurity and included specifics regarding what a solution to the problem of synthetic users might include, simply stated as:

> Design and build a system to create synthetic user and network activity on a network to be used in a customizable and re-playable manner for high-fidelity mission capability and TTP testing. Current systems used throughout the Command and the greater DoD lack the detail needed to simulate real world networks at the fidelity needed to support capability testing and to perform mission rehearsal. The system should be able to collect and anonymize real world network and host data to enable it for re-use in a simulated environment in a configurable manner.
> [USCC 2019]

Similarly, The United States Marine Corps outlines the need for intelligent and realistic user activity in their Intelligent Wargame problem statement:

> Current wargaming tools do not have adaptive nor intelligent adversaries. Whereas computer-driven adversaries indeed do exist, unfortunately, they are driven based on a set of human-defined actions and thus greatly constrained. These human-defined, pre-programmed actions limit the freedom of adversarial engagements and often contain seams that can be easily exploited by users. With this legacy approach, Marine Corps doctrine, strategy, concepts, tactics, techniques, and procedures (TTPs) are not tested against an intelligent adversary capable of employing adaptive and creative tactics that may have never been seen before.
> [USMC 2019]

The point regarding exploitation by users matches our own experience within CERT. Inevitably, there is a high probability of participants “gaming the system” within training or exercise events, given an opportunity. This is not overt cheating, nor is it necessarily an attempt to gain some unfair advantage. Rather, this gaming happens in a myriad of ways — either willing or unknowingly — by taking advantage of game-isms or unrealistic patterns that occur within an

exercise. Simple examples might relate to having limited “shared” internet, where traffic in or out of a friendly network is of unrealistically limited scope. This scenario makes it easy for teams to filter traffic in a way to highlight potential issues very quickly, or pinpoint traffic from specific IP addresses as problematic. In the worst of cases, teams simply whitelist or blacklist IP blocks in a manner that would not work in real-world network operations. This is, of course, just another example of why realism in these events should remain the highest priority for training and exercise builders.

Motivations for Realistic Browsing

Our GHOSTS agents always could browse the internet using any major browser. Agents can be configured to make requests in some particular order or randomly from a supplied list, with the vast majority of implementations utilizing the latter method. It turns out that true randomness is a game-able attribute, however, players using monitoring techniques can infer information about browsing sessions enabling them to filter and unrealistically track sessions.

Our first hint of this problem was our observation of participants tracking the browser’s user-agent (UA) string in a variety of ways while monitoring user-based outbound web requests. The UA uniquely identifies the browser being used, including version, operating system, and type of machine, including laptops, phones, and other computing devices.

Previously we had built mechanisms to change this periodically for any one agent, or even randomize changes to it over time. This included the ability to implement UAs known to be questionable or malicious. We observed teams looking for any UA not following a pattern of being within some recent release of the major browsers. As a result, our use of alternative or malicious strings was immediately flagged. Uncovering the extent to which teams use this piece of information in their filtering and monitoring forced us to rethink the value of true randomization, and led to a complete re-think of what real web browsing behavior looks like on a typical network.

And so, a recent effort within the GHOSTS software platform has been to look at the browsing patterns for agents — what does realistic web browsing look like to a network team? What is the motivation behind that browsing pattern, and how can we implement a solution that introduces the right degree of randomness for a large distributed system, but where it is not so random that it identifies to players as being computer-generated?

As an example, if we sit down and begin browsing the web, what task are we attempting to accomplish? Are we reading the daily news? Are we shopping for new shoes? Or are we using a

[Distribution Statement A] Approved for public release and unlimited distribution.

web application required for our job? If we think of browsing in this fashion, an agent that browses websites in a completely random fashion, going from news, to sports, to shoes, seems entirely out of place.

Further, we wanted to introduce the notion of stickiness to a site — browsing beyond the home page, and properly simulating a person's clicking relevant links on a page within a site.

NPC Context and Preferences

To date within the GHOSTS platform, for every activity an agent executes, we have a full record of what was done and if applicable, what the results were at that time. We might use any or all of this data to make new decisions, but how should agents use that data to make decisions? How might past decisions influence future ones? At this point we might be able to write code to calculate an agent's preferences in making a decision, we had no tally available that might tell us an agent's overall state, what their preference might be between two choices, or what past decisions were, and what the result of that decision was. Examples of this might be that an agent is partial to browsing certain websites, performing some set of tasks, or responding to emails in a particular manner. It might also include having some negative partiality — avoiding certain things or tasks as well.

Although we begin by simply wanting to improve the browsing of relevant links on a website, we introduce a far more reaching capability: This summary state essentially provides an agent with context for making continuous decisions about its future.

Now computational context might be best defined for our purposes by Schmidt et. al. and summarized by Chen as being:

> Context is a description of the situation and the environment a device or a user is in. Schmidt et al. categorized context into six high-level subspaces. Three relate to human factors: information about the user, social environment, and user's tasks. The other three concern the physical environment: Location, infrastructure, and physical conditions.
> [Chen 2005]

We've already given a few examples of user preferences, and we might think of the relationship between the other two, social environment and tasking, as agents being part of a certain team that performs tasks specific to that team. In the past, we've built training and exercise to model team behaviors, such as team A performs this set of specific tasks, and team B does some other separate list — much like you might expect a logistics and marketing team to do so in the

[Distribution Statement A] Approved for public release and unlimited distribution.

corporate world. By assigning these preferences to agents, we can now replicate these team configurations more dynamically and enable them to evolve.

If our agents have preferences, we will want to update them or add new ones over time. We might think of preferences as key/value pairs where the value is some integer. But then, isn't the change of an agent's preferences important? It turns out that having snapshots of context (an agent's preferences) at each decision point gives us the ability to compare decisions between different agents:

> To model context in a CF system, a user's choice or preference needs to be associated with the context in which the user made that choice. This means that the current context needs to be captured each time the user makes a choice. The same applies to the reciprocal: when a user asks for recommendations, we need to capture the current context and evaluate what others have chosen in a similar context in the past. This poses two main problems: how to manage the context in the user profile in terms of data modeling and storage, and how to measure similarities between contexts.
> [Chen 2005]

With this background, our approach to solving the problem of realistic browsing and learning from that activity over time is to use machine learning techniques focused on personalization. Our discussion here focuses on a single feature, however, we have similar agent behaviors within GHOSTS that we believe we can work to understand and improve over time. The user models that different exercises implement via GHOSTS are vast and will only continue to grow, and so this understanding of how agents make decisions will be important guidelines that we can provide to teams as they continue to train and exercise in ever-evolving cyber scenarios.

Agent Personas, Preferences, and Decision-Making

We use the term "preference" here to include that of comparison, prioritization, and choice ranking. So, if preferences are evaluations, they concern matters of value to an agent and are in part, answers to any question about what should be done. They also allow for the comparison of something relative to another, similar thing.

As GHOSTS agents make more informed, and hopefully, more complex decisions, there is a need for each agent to have a system of preferences existing at the time the agent is created, and for an ability to update those preferences over time as the agent continues to make decisions and measure the outcome of those decisions afterward.

To expedite the creation of like agents, the initial preferences of an agent are drawn from pre-defined persona archetypes. Each persona has a set of ranked "interest" attributes, such as a preference for news, sports, or entertainment, for example. These personas are used as

[Distribution Statement A] Approved for public release and unlimited distribution.

baselines for the interests of a newly created agent. The values of the archetype are copied down to the individual agent with a degree of randomness to maintain agent heterogeneity. So where a persona has a range for a given preference, the agent is assigned an initial fixed value. An example of this might be that an enclave of users in logistics is drawn from a persona where there are several applications used to manage logistics tasks for the group. Where the persona has a range for each of these applications, on creation, agents get a random fixed number from that range, and so among individual NPCs in the enclave, some agents "prefer" or use application A over B, and so on. It is important to note that interests are typically multi-faceted, and so it is likely that any single agent will have several interests, and decisions will need to factor this in.

So given an agent has a list of preferences, they might use their values in making forward-looking decisions and update those same preference values based on the eventual outcomes of their choices.

Implementation

Ultimately, we want any single particular agent's browsing history to show patterns where we might say they were reading today's news when they started their shift or shopping for new shoes over lunch or some combination of history items that points to an overarching task. In this case, we believe even a simple pattern reflecting some task would be a step improvement over purely random browsing.

Purely random browsing has been a common use-case for GHOSTS, but the problem for us has been that truly random browsing does not match human behavior, where we are looking for some piece of information or looking to solve some specific task. Pure random results in a browser history that bounces from site to site with no apparent connections between, as if an agent had no intent behind their browsing activity.

In an attempt to move away from sheer arbitrariness, our first step is to categorize all of the URLs an agent visits, so that each site is a member of *n+1* categories. Of course, on-the-fly categorization of a given web URL is a current machine learning problem, but sufficient progress has been made that we might leverage the output of those that have gone before us to solve our problem. Since we control the internet in any simulation, training, or exercise event, we can pre-classify all potential websites that any agent might browse.

[Distribution Statement A] Approved for public release and unlimited distribution.

Classifying Websites

Classifying websites is our first step. For a list of top websites, we used the Majestic Million[3], which claims to be one of the world's largest commercially available link databases. This provided us with information such as the site, its global rank in terms of traffic, and other data not necessary for our purposes:

Rank (as of Q1 2020)	Site
1	Google.com
2	Facebook.com
3	Youtube.com
4	Twitter.com
5	Instagram.com
6	Linkedin.com
7	Microsoft.com
8	Apple.com
9	Wikipedia.org
10	Plus.google.com

Next, we need to categorize websites with the same listings we would use in defining interests for our NPCs. A simple way to think about categorization for our purposes would be how a web directory might list a particular site. Although the use of these sites is perhaps waning in the age of search, they still exist, and DMOZ[4] fits our need to at least have a single category for each site in our listing.

To be able to cross-reference our list of domains with a category, we polled each site and captured relevant meta-data, including the site's keywords and description to cross-reference with our selected NPC categories. This was done by simple keyword matching that we had built for our NPC categories enabled us to cross-reference sites with categories and tag each appropriately:

Site	Site Description	Keywords	Category
http://wikipedia.org	Wikipedia is a free online encyclopedia, created and edited by volunteers around the world and hosted by the Wikimedia Foundation	encyclopedia, dictionary, learning	Education
http://craigslist.org	craigslist provides local classifieds and forums for jobs, housing, for sale, services, local community, and events	sale, shopping, deals	Shopping

[3] https://majestic.com/reports/majestic-million
[4] https://www.dmoz-odp.org/

[Distribution Statement A] Approved for public release and unlimited distribution.

http://reddit.com	Reddit is a network of communities based on people's interests. Find communities you're interested in, and become part of an online community!	community, interest, hobby	Community
http://imdb.com	IMDb is the world's most popular and authoritative source for movie, TV, and celebrity content. Find ratings and reviews for the newest movie and TV shows	movies, tv, music, culture	Entertainment
http://buzzfeed.com	BuzzFeed has breaking news, vital journalism, quizzes, videos, celeb news, Tasty food videos, recipes, DIY hacks, and all the trending buzz you'll want to share with your friends	news, journalism	News

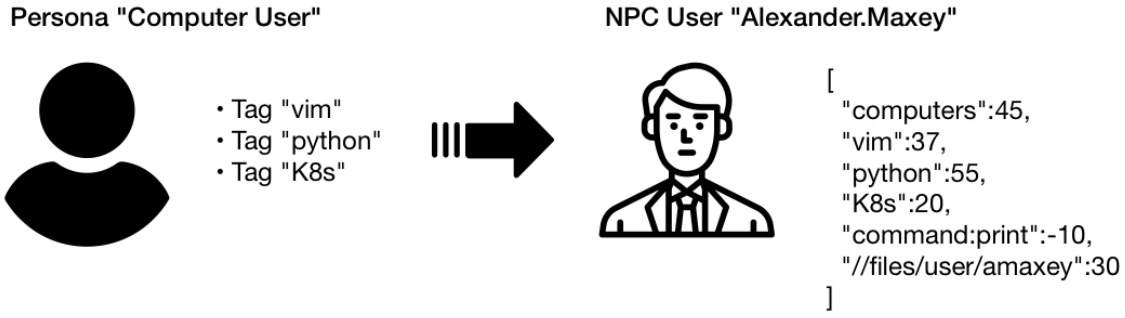
While our approach is one of many methods one might use to generate a categorized listing of sites, it hopefully motivates our intent and provides a "how possibly" explanation of doing so.

NPC Preference Engine

The next thing our GHOSTS agents would need is some preference for which site to browse next, but we want to keep in mind that a more general preference engine will be helpful in the future for other decisions agents might want to make. To best facilitate our needs, and provide room for growth in the future, we implemented a simple key-value pair of the preference and an integer ranging from 100, representing a particularly strong dislike, to 100, representing a strong preference for that thing. So an agent with a strong preference for apples and a dislike for oranges we would represent as:

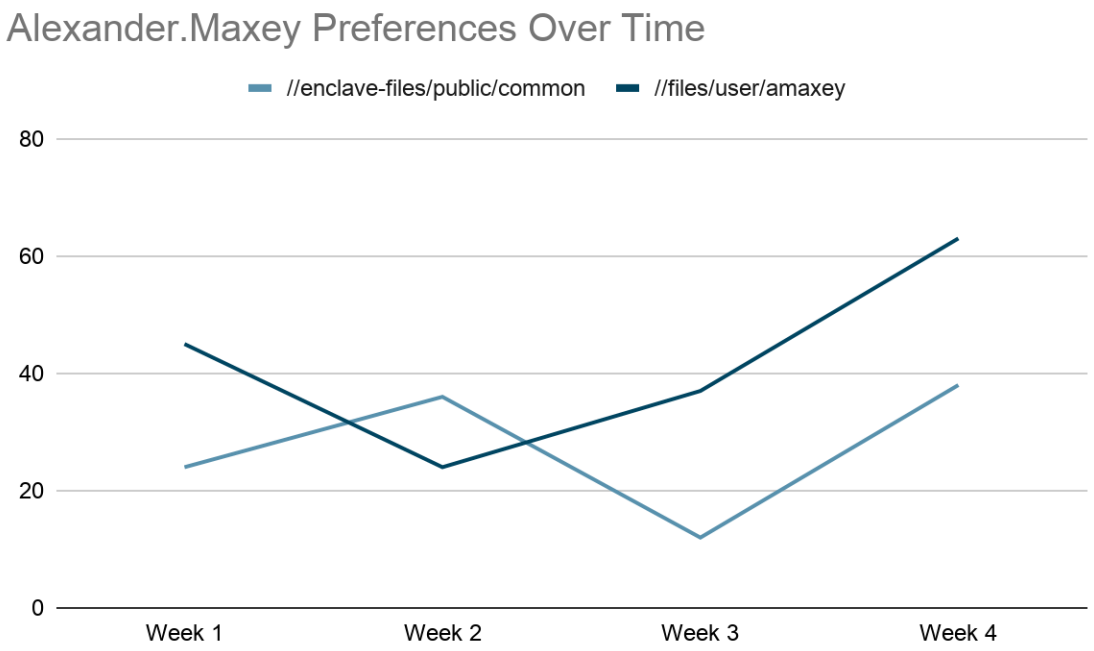
```
[{"apples":100}, {"oranges":-100}]
```

Of course, this array can contain any number of preferences, and while we can have general preferences like "apples", we can be more or less precise as needed. For example:



[Distribution Statement A] Approved for public release and unlimited distribution.

Of course, users can collect new preferences, and their existing preferences can change over time. These changes are handled transactionally, so increases or decreases in a particular preference are tracked, and we could go back to any point in time and determine what an agent's preference was at the time and how it has changed since. As an example, a user might change where they are storing network files over time as their job or role within an organization changes:



Now that we have NPCs that prefer to do some things over others, we can look more closely at what tasks they might perform within the browser, and how they might browse to complete that task. We can also have NPCs browse for information over lunch aligned to their preferences as well so that sports fans might find out more about the latest soccer scores, and the like.

Building and Applying an NPC Preference Model

To build a machine learning model for improving NPC browsing patterns to more closely match their browser site history to their preferences, we will need three sets of data: NPC preferences, their current browser history, and our list of sites that we categorized. With this data, the question we will ask is, does your browser history thus far match what content we would expect you to be looking at, given your role or your likes and dislikes?

[Distribution Statement A] Approved for public release and unlimited distribution.

As we discussed in previous sections, we have a generated list of sites and their classifications based on their content, and we have a mechanism for assigning a persona to an NPC and getting applicable particular preference settings. Since the detailed history of what every GHOSTS agent does gets logged and collected, we can use that to reconstruct any single user's browsing history.

We approached building a machine learning model to provide better browsing patterns in the same spirit as to how shopping sites attempt to recommend products that you might be most interested in based on your prior activity or some other datapoint. An example would be if you're considering buying a new laptop, perhaps you would be also interested in an extra charger as well? Another might be that based on your prior purchase of a laptop, perhaps you would now be interested in another charger?

Our work follows this latter case — based on sites that you have browsed in the past, and its alignment to your preferences, would you browse it in the future? And if so, would you be interested in other sites (and what might they be?) similar to this one? In the same spirit of having a purchase history, we have an existing browsing history for each NPC. We can take this history for each NPC and:

1. Determine if the site matches any NPC preference, either positive or negative,
2. Based on this, add or remove the site from the next iteration of sites to browse,
3. Based on the final aggregate of sites the NPC is interested in, find sites similar to this set.

Step (3) will incorporate our machine learning model that finds sites similar to the NPC's preferences after an iteration of browsing. At this point, let's look at the model classes necessary for this evaluation:

Class	**Parameter**	**Description**
Web Site	Id	Site identifier cmu.edu
	Categories	Set of site categories ["education", "university"]
NPC Agent	Id	NPC identifier
	Preferences	Set of preference values [{"education":20}, {"sports":10}]
Browse Prediction	NPC Agent Id	Id for NPC
	Web Site Id	Id for site
	Score	Likelihood of this site matching NPC's preferences

[Distribution Statement A] Approved for public release and unlimited distribution.

The model will then provide us with a likelihood match for each potential new site matching an NPC's past browsing history. We can make an arbitrary determination for what constitutes a strong correlation here, but perhaps above .5 (50%) is accurate enough for our purposes, and perhaps allows for some new additions to enter the NPC's preference list in the future. Since we want to organically grow the list of sites that the NPC browses often, a lower likelihood threshold also allows us to avoid overfitting the model and coming to a constrained set of sites too early as well. This overfitting will be a continued problem in this space where ML is used to tailor NPC activity. As we want to allow for the randomness that humans sometimes exhibit in NPC activity, we want to be careful to allow this to remain regardless of how many times a model might be run.

Results

For our models, we created 25 NPCs and as part of that generation, assigned them one of 13 primary personas: Arts, Business, Computers, Games, Health, Home, Kids, News, Recreation, Reference, Science, Shopping, and Society. Each NPC was assigned a blend of unique preferences based on that persona with strengths from 0 to 100 for indifferent or no preference to the strongest of preference. To keep things simple for our purposes here, we excluded negative preferences, (which would have been allowing the strength value a range of -100 to 100, to include strong negative preference) and so, we do not account for NPCs avoiding certain sites because they do not prefer them. Rather, we focus solely on scenarios where a site has content relevant to something an NPC is interested in (has a preference for).

We track two values for each NPC:

- *Start* is the original percent of sites in the browsing history that match the current preferences.

- *End* is the percent of sites in the browsing history matching preferences after adjusting for the recommendations by our model.

- *Gain* is the difference between the end and start values. A positive value here would indicate that the browsing list is more tightly aligned with an agent's preferences once run through the model.

Although we ran many interim tests, the contrast between three particular models illustrates the progress we made in tuning the model and shows what's possible in this space of improving NPC realism. These tests are best described as:

[Distribution Statement A] Approved for public release and unlimited distribution.

1. The Initial Model — M0 (“Random”)
   Our first model included no NPC preferences, and so the results are likely to match simply choosing websites at random for an NPC to browse. This is how most browsing patterns are generated today, so this baseline would help establish the model’s results. Incredibly the average gain for this model is 0% — meaning that the model generates new browsing lists that are no better or worse than picking sites at random.

2. The Middle Model — M1 (“No Negative Feedback”)
   Along the way, we had composed a model where no negative feedback was incorporated into the model. Instead, we discarded suspect matches, removing them from the training data entirely, and fed only positive match examples. With the model not having any negative results to balance its choices, predictably our results were worse than picking at random.

3. The Final Model — M2 (“Balanced”)
   Taking into account what we learned from the previous models, our adjustments accounted for a 26% improvement in an NPC’s browsed sites more closely matching their preferences at that time.

			M0 (“Random”)			M1 (“No Negative Feedback”)			M2 (“Balanced”)		
NPC	**Primary Preference**	**Preference Strength**	**Start**	**Final**	**Gain**	**Start**	**Final**	**Gain**	**Start**	**Final**	**Gain**
Alexander.Maxey	Computers	44.5	79%	93%	14%	79%	92%	13%	79%	99%	20%
Alfredo.Seaman	Kids	44	63%	41%	-22%	63%	27%	-36%	63%	96%	33%
Allan.Deal	Recreation	32.5	68%	64%	-4%	68%	49%	-18%	68%	98%	30%
Ashley.Munson	Arts	37	70%	88%	18%	70%	79%	9%	70%	96%	26%
Carissa.Kelso	Reference	37.5	74%	78%	4%	74%	49%	-25%	74%	98%	24%
Cecelia.Nunley	Society	43	79%	87%	8%	79%	37%	-42%	79%	98%	20%
Clinton.Belt	Recreation	36.5	68%	70%	2%	68%	41%	-28%	68%	98%	29%
Conor.Rouse	Reference	52	80%	78%	-2%	80%	63%	-17%	80%	98%	19%
Dominick.Ragan	News	56	75%	43%	-32%	75%	39%	-37%	75%	98%	23%
Donte.Gillette	Home	43.5	63%	35%	-29%	63%	31%	-33%	63%	96%	32%
Emmanuel.Battle	Games	14	26%	45%	19%	26%	20%	-6%	26%	81%	55%
Jaron.Lindstrom	Computers	44	80%	93%	13%	80%	93%	13%	80%	99%	19%
Joey.Crowder	Business	35.5	64%	84%	20%	64%	86%	22%	64%	96%	32%
Joseph.Mosley	Reference	31	72%	75%	4%	72%	66%	-6%	72%	98%	26%
Kacy.Kinder	News	39	68%	50%	-18%	68%	21%	-47%	68%	97%	29%
Krystal.Shepherd	Arts	40.5	71%	87%	16%	71%	81%	10%	71%	98%	27%
Lana.Girard	Society	16	67%	83%	16%	67%	26%	-41%	67%	89%	22%
Laurie.Fleming	Shopping	40.5	69%	58%	-11%	69%	70%	1%	69%	97%	28%
Leslie.Richmond	Society	42.5	77%	84%	7%	77%	69%	-8%	77%	98%	21%

[Distribution Statement A] Approved for public release and unlimited distribution.

Racheal.Denney	Computers	37	78%	93%	15%	78%	91%	13%	78%	98%	21%
Rashawn.Dow	Science	54	80%	58%	-21%	80%	72%	-7%	80%	99%	19%
Rodrigo.Rojas	Recreation	32.5	68%	63%	-5%	68%	18%	-50%	68%	98%	30%
Shayne.Fraley	Science	60	81%	62%	-19%	81%	23%	-58%	81%	99%	17%
Tarah.Meredith	Shopping	37.5	68%	74%	6%	68%	65%	-3%	68%	97%	30%
Tracie.Gamboa	Society	44	78%	87%	9%	78%	34%	-44%	78%	99%	21%
Average Gain:					**0%**			**-17%**			**26%**

Of course, we expect continued improvement with each repetition, with the results of an iteration feeding the next.

Conclusions

In the quest for continued realism in cybersecurity exercise and training, ML continues to have a unique opportunity to contribute to both the creation of exercise scenarios and artifacts and to also process the data that is created as a result of the event. We show here but one of the ways ML can be used to increase event fidelity and prevent participating teams from “game-isms” and a reduction of value in the exercise. Certainly, we should be mindful of the potential in these models for eventual overfit, but the value of ML otherwise is too powerful to ignore.

[Distribution Statement A] Approved for public release and unlimited distribution.

References

Chen, Annie. "Context-aware collaborative filtering system: predicting the user's preferences in ubiquitous computing." CHI'05 Extended Abstracts on Human Factors in Computing Systems. ACM, 2005.

Dobson, Geoffrey B., et al. R-EACTR: A Framework for Designing Realistic Cyber Warfare Exercises. No. CMU/SEI-2017-TR-005. CARNEGIE-MELLON UNIV PITTSBURGH PA PITTSBURGH United States, 2017.

Hammerstein, Josh, and Christopher May. The CERT approach to cybersecurity workforce development. No. CMU/SEI-2010-TR-045. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2010.

Schmidt, Albrecht, Michael Beigl, and Hans-W. Gellersen. "There is more to context than location." Computers & Graphics 23.6 (1999): 893-901.

Updyke, Dustin D., et al. Ghosts in the Machine: A Framework for Cyber-Warfare Exercise NPC Simulation. No. CMU/SEI-2018-TR-005. CARNEGIE-MELLON UNIV PITTSBURGH PA PITTSBURGH United States, 2018.

U.S. Cyber Command Technical Challenge Problems Guidance. USCC-J9-TO-2019-03-12. 2019.

United States Marine Corps. Intelligent Wargame (i-Wargame). USMC AI MVP Project Submission. 2019.

[Distribution Statement A] Approved for public release and unlimited distribution.