Carnegie Mellon University
Software Engineering Institute

Using Machine Learning to Increase NPC Fidelity

Dustin D. Updyke
Thomas G. Podnar
Geoffrey B. Dobson
John W. Yarger

December 2021

TECHNICAL REPORT
CMU/SEI-2021-TR-005
DOI: 10.1184/R1/14107373

CERT DIVISION

http://www.sei.cmu.edu

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.

This report was prepared for the SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100

DM21-0167

Table of Contents

List of Figures

List of Tables

Abstract

Experiences that seem real to players in training and exercise scenarios enhance learning. Improving the fidelity of automated non-player characters (NPCs) can increase the level of realism felt by players. In this report, we describe how we used machine learning (ML) modeling to create decision-making preferences for NPCs. In our research, we test ML solutions and confirm that NPCs can exhibit lifelike computer activity that improves over time.

1 Introduction

In this report, we summarize our experiences using machine learning (ML) modeling to create decision-making preferences for non-player characters (NPCs) within a distributed system that supports cybersecurity training and exercise events.[1] We also provide details about how we explored ML solutions to ultimately arrive at one that enables NPCs to exhibit lifelike computer activity that becomes even more lifelike over time.

The terms *NPC* and *agent* are central to using ML to create experiences that seem real to players in training and exercise scenarios.[2] An NPC is the personality or character within the scenario. An agent is a software instance that is configured to bring the NPC "to life" as a character in the scenario.[3]

NPCs simulate real-world user activity and create accurate network traffic. We maintain that the best cyber-defense teams triangulate their findings based on network traffic, logs, sensor data, and a growing host-based toolchain. Therefore, we focus on overall activity realism and confirm our approach by comparing NPC activity to real-world users performing the same activity.

We describe two common problems present in many current ML implementations. We resolve those problems by implementing ML models that use personas and preferences. We hope that simulation designers will consider using our solutions and/or our general approach. Furthermore, anyone interested in building an introductory implementation of ML may want to use our solution since it does not require a significant investment of money, resources, or data.

We use GHOSTS[1] in our research, but the concepts throughout this technical report could be adapted to other NPC frameworks as well.

1.1 The CERT CWD Team

For over a decade, our CERT Cyber Workforce Development (CWD) team has continually highlighted the need for realism within elite cybersecurity training and exercise events. Our approach to constructing and executing these events builds the knowledge, skill, and abilities of players and teams in a measurable and repeatable fashion [Hammerstein 2010].

---

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

2 We chose to use the term *NPC* to span many other terms in the synthetic-user domain, such as *user sim* and *network traffic,* which point to the desire for high-fidelity, lifelike activity on a computer network.

3 "[A]n agent is the raw binary client executing commands on a machine, the layer above it brings the NPC "to life" by infusing it with human-like characteristics, beliefs, and intentions" [Updyke 2018].

Based on our experience executing hundreds of high-fidelity events, we created the following:

- **R-EACTR Framework.** The *Realistic - Environment, Adversary, Communications, Tactics, and Roles (R-EACTR)* design framework establishes a method for building cyber-warfare exercises. It contains details about realism and its impact within exercise events [Dobson 2017].
- **GHOSTS NPC Orchestration Framework.** The GHOSTS framework establishes a method for building behaviorally accurate, autonomous NPCs within high-fidelity cybersecurity training and exercises [Updyke 2018]. This framework replicates the behavior of many different types of real-world users on an enterprise network—including blue and red teams. GHOSTS NPCs use operating system applications to generate network traffic.
- **GHOSTS Software Project.** To demonstrate and further develop GHOSTS framework concepts, we released a modular, open source software project called GHOSTS.[4] With GHOSTS, cybersecurity experts can test their skills and realistically train, defending real networks containing NPCs doing the things we expect them to do (e.g., create documents, access systems, browse the web, run commands). We continue to actively develop this software. GHOSTS is also supported by the open-source training and exercise community, which contributes to its growth and expansion.

1.2 Challenges

There is value in providing players with realistic, high-fidelity training and exercise scenarios. ML holds the key to building a "thinking" teammate or adversary in these scenarios. However, there are still some challenges to consider:

- **Fidelity of User Simulations.** The need for increasing the fidelity of user simulation is not new. It remains a priority to the military. In 2019, the United States Cyber Command released challenge guidance for 39 critical areas in cybersecurity. The guidance for the synthetic user challenge included

  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 (USMC) Training and Education Command (TECOM) outlined the need for intelligent and realistic synthetic 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

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

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]

- **Gaming the System**. We can validate the challenge articulated by TECOM. Players are always looking for patterns and will quickly exploit NPC weaknesses. This "gaming of the system" is not cheating nor is it an attempt to gain an unfair advantage. Rather, it happens in different ways—either knowingly or unknowingly—by leveraging *game-isms* (i.e., unrealistic patterns that occur in an exercise).

  An example of a game-ism is when an exercise offers a limited shared Internet, where the scope of traffic in or out of a friendly network is unrealistically limited. This scenario makes it easy for players to (1) filter traffic to highlight potential issues quickly or (2) identify traffic from specific IP addresses as problematic. In the worst case, players can place IP blocks in approved or unapproved lists—a method that would not work in real-world network operations. This example underscores why realism should remain the highest priority for training and exercise builders.

1.3 About This Report

Ultimately, cybersecurity training and exercises will require the coordination of distributed software agents that drive NPCs and their activities. This automation is only available using ML. The fidelity of user simulations will continue to be a challenge. Users will continue gaming the system. To tackle these challenges, we turn to ML. In this report, we describe our efforts to

- deliver realistic NPC web browsing
- understand and improve NPC decision making by studying NPC context and preferences
- use personas to influence NPC preferences and decision making
- implement ML models that include preferences and decision making

2 Delivering Realistic Browsing

To improve the fidelity of user simulation, our GHOSTS software agents enable NPCs to browse the Internet using any major browser. We configure agents to associate NPCs with preferences by making requests in a particular order or randomly using a supplied list; most implementations use the latter method. True randomness is a "game-able" (i.e., exploitable) attribute.

Players using monitoring techniques can infer information about browsing sessions, and these inferences enable them to filter and unrealistically track sessions. Our first hint of this problem was when we observed players tracking the NPC browser's user-agent (UA) string in different ways while monitoring NPC-based outbound web requests. The UA string uniquely identifies the browser being used, including its version, operating system, and type of machine (e.g., laptops, phones, and other computing devices), as shown in Figure 1. Previously, we built mechanisms to change this UA string periodically for each NPC, or even randomize changes to it over time.[5] With this approach, we can also implement UA strings known to be questionable or malicious. However, we observed players "gaming the system" by looking for UA strings that didn't follow the patterns of UA strings in recent releases of major browsers. As a result, players flagged our use of alternative or malicious strings immediately.

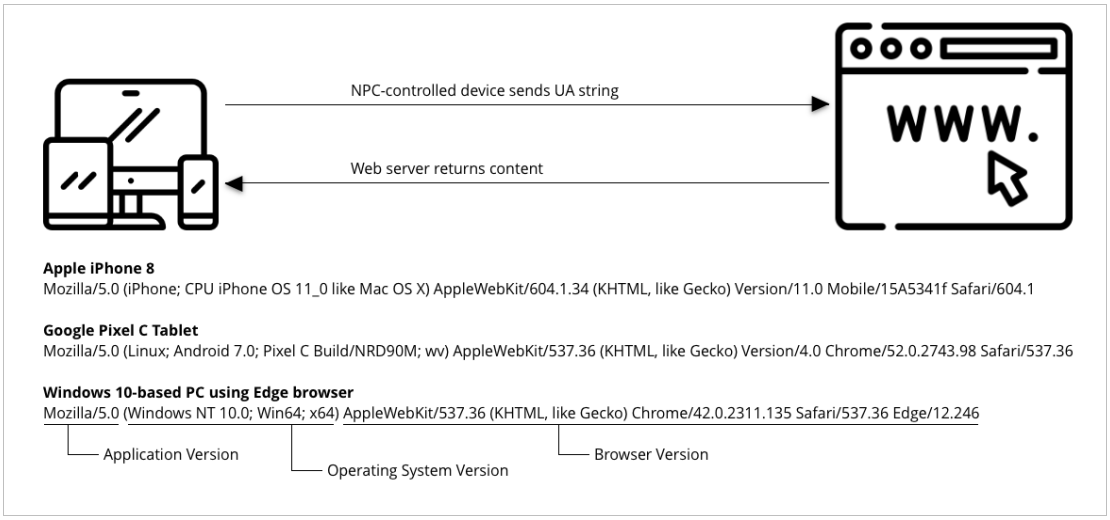

Figure 1: Example User-Agent Strings and Their Use in Browsing

The extent to which player teams used this information in their filtering and monitoring forced us to rethink the value of true randomization and to re-examine what real-world browsing behavior looks like on a typical network.

[5] Changing the UA string simulates how a user might update or change their web browser periodically over time.

2.1 Real-World Browsing Patterns

We used the GHOSTS NPC Orchestration Framework to examine patterns in NPC browsing behavior and asked questions such as

- What does realistic web browsing look like to a network team?
- What is the motivation behind particular browsing patterns?
- In a large distributed system, how can we introduce the right degree of randomness without alerting players that the randomness is computer generated?

When researching browsing patterns, we thought about what people do when browsing the web. What is their goal? Are they reading the news or sports scores? Are they shopping for something? Are they using a web application required for their job? When thinking about browsing in this way, an NPC that browses websites randomly—going from news, to sports, to shopping—seems artificial and inconsistent with the real world.

2.2 Website Stickiness

People often explore a website in depth. They may engage in reading long-form content that is not captured on a single page. They may search through long lists of content that is paginated by design due to its length. They may compare several different items that are showcased in detail on separate pages. They may read news articles that highlight their varied interests. As a result, we introduced the notion of a website's "stickiness" (i.e., an enticement to browse beyond the home page). We implemented this configurable feature with some degree of randomness but also with the ability to have NPCs visit at least some number of additional pages from the page first visited within a site. Once we incorporated stickiness into our approach, we were better able to simulate a user clicking relevant links on pages across a website, thereby increasing the fidelity of NPCs and the agents that control them.

3 Studying NPC Context and Preferences

GHOSTS records every activity a software agent executes to control an NPC. Therefore, there is a complete record of what was done and, if applicable, what the results were at any particular time. How could agents use that data to help NPC make decisions? How could past NPC decisions affect future ones?

For software agents to calculate an NPC's preferences when making a decision, they need access to information representing the following:

- an NPC's overall state
- its preference between available choices
- its past decisions
- the result of its past decisions

Examples of an NPC's preferences are certain websites, particular tasks, and how it responds to emails. Preferences might also include some negative partiality (i.e., avoiding certain tasks).

Our primary goal is to improve how an NPC browses relevant links on a website. However, we also introduce a more ambitious capability: providing context for an NPC to make continuous decisions about its future. For our purposes, this context (i.e., computational context) is best defined by Schmidt et al. and summarized by Chen [Chen 2005]:

> *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.*

We already provided a few examples of NPC (our "user") preferences, and we might think of the relationship between the other two (i.e., social environment and tasking) as NPCs that are part of a team that performs tasks specific to that team. In the past, we built training and exercises to model real-world team behaviors. For example, *Team A performs this set of specific tasks*, and *Team B performs some other separate set of tasks* (much like you might expect a logistics and marketing team to do in the corporate world). By assigning these preferences to NPCs, we replicate these team configurations more dynamically and enable them to evolve.

We think of preferences as key/value pairs where the value is an integer. If our NPCs have preferences, we should update them or add new ones over time. For research purposes, having a transactional and historical record of how and when an NPC changes its preferences provides valuable data. Annie Chen supports this when describing how having snapshots of context (i.e., an NPC's preferences) at each decision point enables us to compare decisions among different NPCs [Chen 2005]:

> *To model context in a [collaborative filtering] 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.*

Our approach to solving the challenge of realistic browsing and learning from the context and decisions the NPCs make over time is to use ML techniques that focus on personalization. In this report, we focus on this single feature; however, there are similar NPC behaviors in GHOSTS that can help us understand and improve these behaviors over time. The user models[6] that are implemented in different exercises via GHOSTS are vast and will continue to grow; therefore, understanding how NPCs make decisions provides important guidelines to help player teams as they train and perform exercises in ever-evolving cyber scenarios.

[6] User models are representations of types of users and can range from friendly administrators to hostile nation state actors.

4 Using Personas to Influence NPC Preferences and Decision Making

In this report, the term *preference* includes comparison, prioritization, and choice ranking. Therefore, if preferences are evaluations, they are valuable to an NPC and provide context to help inform decisions. Preferences also enable an NPC to compare similar things.

As GHOSTS NPCs make more informed and more complex decisions, there is a need for each NPC to (1) have an existing system of preferences when it is created and (2) be able to update those preferences over time as it makes decisions and measures the outcomes.

To expedite creating NPCs with similar capabilities, the initial preferences are drawn from a predefined *persona*. Each persona has a set of ranked interest attributes, such as a preference for news, sports, or entertainment. To maintain an NPC's heterogeneity, the values of a persona are copied to the individual NPCs randomly. So, when a persona has a range for a given preference, an NPC is assigned to an initial fixed value.

For example, an enclave of NPCs in logistics is drawn from a persona with several applications used to manage logistics tasks. The persona has a range for each of these applications; when agents are created, they get a random fixed number from that range. Therefore, among individual NPCs in the enclave, some "prefer" application A over B. Interests are typically multi-faceted, so a single NPC can have several interests; decisions must account for these multiple interests.

5 Implementing ML Models That Include Preferences and Decision Making

The goal is for a particular NPC's browsing history to show patterns that reflect its activities (e.g., reading the news when the NPC starts its shift or shops for new shoes over lunch). Examining a browsing history should identify overarching tasks. In this case, even a simple pattern that reflects a task is an improvement over purely random browsing.

Purely random browsing was a simple, common use case for most user simulations, but this approach doesn't mirror human behavior. In human behavior, we can look for specific information or execute a specific task. But purely random browsing produces a browser history that bounces from site to site arbitrarily—with no apparent connections or reason, as though the NPC has no intent behind its browsing activities.

To shift from this arbitrariness, we (1) categorize all the websites an NPC visits and (2) build and apply a preference engine.

5.1 Classifying Websites

Classifying the websites that an NPC agent could visit should result in each website being a member of $n+1$ categories. This type of categorization is an ML problem, and ML researchers are continually refining many different approaches to its solution.[7]

Since we control the Internet in any simulation, training, or exercise event, we can pre-classify all websites that an NPC might browse. To do this, we created a list of top sites and categorized them.[8]

Produce a List of Websites

We used Majestic Million to produce a list of top websites.[9] Majestic Million provided the type of data we needed, such as the site and its global ranking based on traffic volume. Table 1 lists the top 10 sites according to Majestic Million in the first quarter of 2020.

---

7 Google Scholar currently lists about 119,000 results for the search term "website categorization with machine learning."

8 Many of our training and exercise scenarios implement CWD's TopGen (https://github.com/cmu-sei/topgen) for simulating websites. This open-source virtualized service simulator allows a single host (e.g., physical, virtual machine [VM], or container) to serve multiple co-hosted virtual services (such as multiple HTTP vhosts, DNS views, and/or SMTP/IMAP virtual mail domains).

9 Majestic Million surveys and maps the Internet and has created one of the largest commercial link intelligence databases in the world (https://majestic.com/reports/majestic-million).

Table 1: Majestic Million's Top Ten Websites from Q1 2020

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

Categorize the Websites

Next, we categorized the websites with the same attributes we use to define interests for our NPCs. A simple way to think about categorization is to consider how a web directory might list a particular site. Because web searches have become ubiquitous, web directories aren't as broadly used, but they still exist. For our purposes, DMOZ[10] is useful because it offers at least a single category for each site in our listing. (See Table 2 and Table 3.)

[10] DMOZ, short for directory.mozilla.org, is an open-content directory of links on the World Wide Web (https://www.dmoz-odp.org/).

Table 2: Categories Used to Classify Websites

Categories
Arts
Business
Computers
Games
Health
Home
Kids
News
Recreation
Reference
Science
Shopping
Society

By cross-referencing our list of domains with a category, we could align NPC browsing to the sites that match their preferences. We polled each site and captured relevant metadata—including the site's keywords and description to cross-reference that information with our selected NPC categories. We did this by performing simple keyword matching for the keywords we previously built for our NPC categories. This enabled us to cross-reference sites with categories and tag each one appropriately. See Table 3.

Table 3: Websites Annotated with Descriptions, Keywords, and Categories

Site	Site Description	Keywords	Category
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
craigslist.org	craigslist provides local classifieds and forums for jobs, housing, for sale, services, local community, and events.	sale, shopping, deals	Shopping
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
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
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

Our approach is one of many methods used to generate a categorized list of sites.

5.2 NPC Preference Engine[11]

Our GHOSTS NPCs need a preference that "motivates" them to select which site to browse next. In the future, a more general preference engine will help form other decisions that NPCs might want to make. For now, and to provide room for growth, we represented each preference with a simple key/value pair. Keys can be any unique string. Values must be an integer ranging from 100 (representing a strong preference) to -100 (representing a particularly strong dislike). Using this approach, an NPC with a strong preference for computers and a strong dislike for printing would be represented as

```
[{"computers":100}, {"printing":-100}]
```

An NPC can have any number of preferences, and while they can have general preferences like "computers," that preference can also be far more precise, perhaps indicating a specific preferred software application, printer, or file share. See Figure 2 for an example.

[11] The NPC Preference Engine is a part of SPECTRE (https://github.com/cmu-sei/GHOSTS-SPECTRE), an optional package within the GHOSTS framework. It currently runs as a separate application container alongside the standard GHOSTS API container.

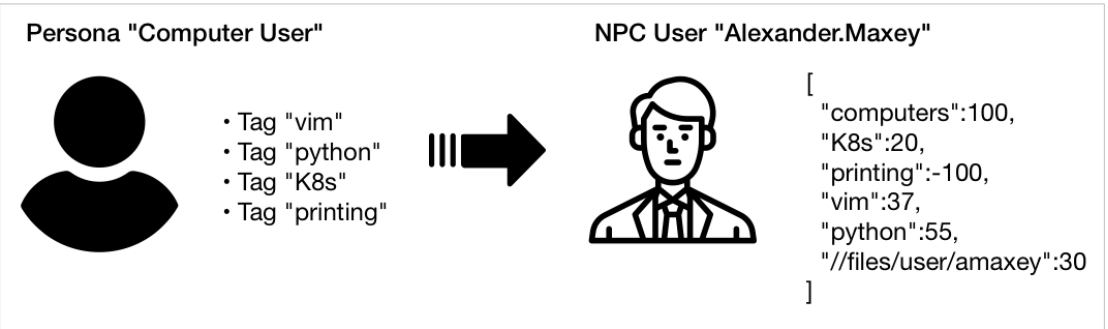

Figure 2: Precision in Preferences

NPCs 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, so we could go back to any point in time and determine what an NPC's preference was and how it has changed. For example, a user might change where they store network files over time as their role in an organization changes, as illustrated in Figure 3.

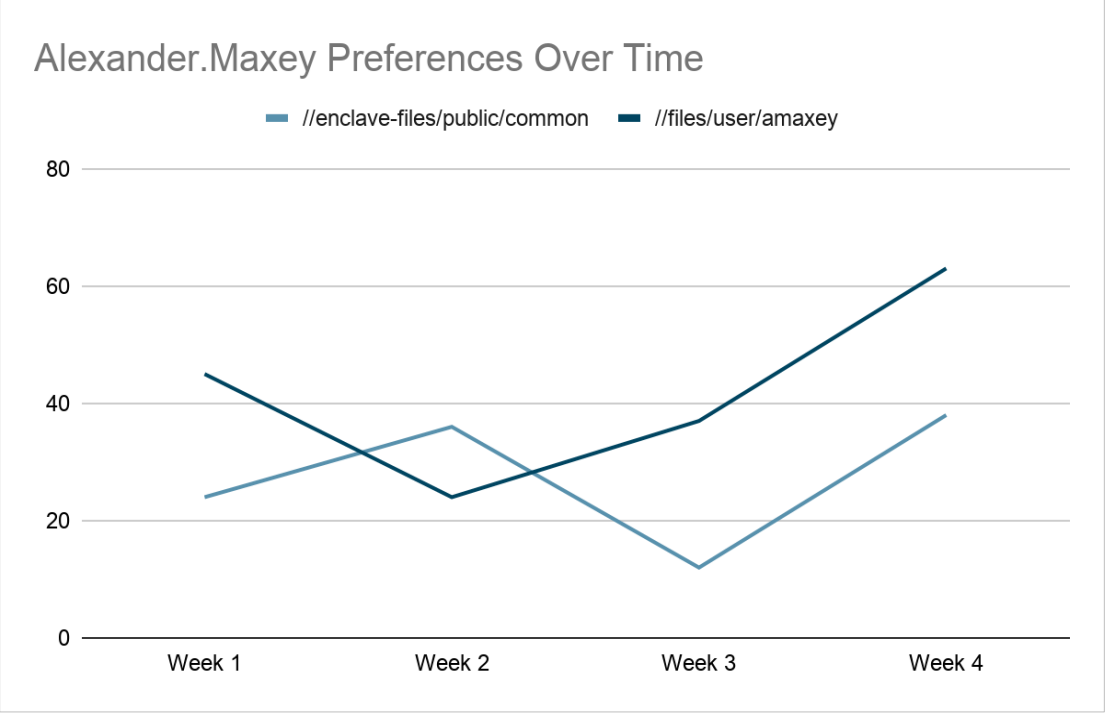

Figure 3: File-Storing Preferences Over Time

Now that we have NPCs that prefer to do some things over others, we can look more closely at the tasks they might perform from a browser and how they might browse to complete that task. We can also align an NPC's preferences to browse for information over lunch so that sports fans can get the latest scores.

5.3 Building and Applying an NPC Preference Model

To accomplish our goal of building an ML model that improves NPC browsing patterns in a way that more closely matches its browsing history to its preferences, we need three sets of data:

- NPC preferences
- current NPC browser history
- list of categorized websites

With this data, we might consider each NPC in terms of the question, "Does your browser history match the content associated with your role and preferences?"

As discussed in previous sections, we have

- a list of websites and their classifications based on their content
- a mechanism for assigning a persona to an NPC and acquiring the applicable preference settings

Since the detailed history of every GHOSTS NPC's action is logged, we can reconstruct any single NPC's browsing history.

Remember that our goal is to build an ML model that provides better browsing patterns. We approached this task in the same way consumer sites use data (e.g., using a shopper's prior activity or purchase history to recommend products that might interest them). If a shopper is looking for a new laptop, the consumer site might ask them if they are interested in buying an extra laptop charger as well.[12]

In our ML model, we ask the NPC these questions:

- Based on (1) sites that you have browsed in the past and (2) a site's alignment to your preferences, would you browse this site in the future?
- If yes, would you be interested browsing other sites?
- What might those sites be?
- Are those sites similar to this one?

Similar to consumers having a purchase history, we have an NPC's browsing history. Using browsing history, we can perform the following steps:

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

[12] Our initial direction was partly driven by the wide availability of ML-driven personalized product recommendation engines. By substituting websites for products, we were able to build several rough proof-of-concept models that encouraged us to further pursue the research outlined in this report.

Step 3 incorporates our ML model, which finds sites similar to the NPC's preferences after an iteration of browsing. Table 4 depicts the model classes necessary for this evaluation.

Table 4: Necessary Model Classes

Class	Parameter	Description
Website	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
	Website Id	Id for site
	Score	Degree to which this site matches NPC's preferences

The model assigns a score to each new site that reflects the degree to which that site will match the NPC's preferences. We can make an arbitrary determination of what constitutes a strong correlation, but a score above .5 (50) may be accurate enough for our purposes, and it allows future additions to the NPC's preferences.

Since we want to organically grow the list of sites that the NPC browses often, a lower likelihood threshold allows us to avoid (1) overfitting the model and (2) constraining the set of sites too early. Model overfitting is an ongoing problem when using ML to tailor NPC activity. NPC activity should also reflect the randomness that humans sometimes exhibit.[13] Therefore, we must be careful to allow this type of randomness regardless of how many times the model is run.

[13] NPC activity could also be influenced by scenario planners as they design events that lead NPCs to browse particular categories, such as a news-worthy event or a new order from higher headquarters.

6 Results

We created 25 NPCs for our model and assigned each of them 1 of the following 13 primary personas:[14]

1. Arts
2. Business
3. Computers
4. Games
5. Health
6. Home
7. Kids
8. News
9. Recreation
10. Reference
11. Science
12. Shopping
13. Society

A blend of unique preferences was assigned to each NPC based on the persona. Each preference was associated with strengths from 0 (indifferent or no preference) to 100 (very strong preference). To simplify our initial model and results, we excluded negative preferences (i.e., allowing the strength value a range of -100 to 100, including strong negative preference). Therefore, we do not account for NPCs avoiding certain sites simply because they do not prefer them. Instead, we focused on scenarios to select sites that had content relevant to what an NPC preferred.

We tracked three values for each NPC:

- *Start* is the original percent of sites in the browsing history that match the NPC's current preferences.
- *End* is the percent of sites in the browsing history that match the NPC's preferences after adjusting for the recommendations by the particular ML model.
- *Gain* is the difference between the end and start values. A positive gain indicates that the browsing list is more closely aligned with the NPC's preferences once the ML model is applied.

[14] We chose the 13 personas and the preferences they contain arbitrarily, assigned 1 persona per NPC, and aligned these to our website classifications for simplicity. Future work could greatly expand the number of possible personas or combine sets of personas per NPC. Correlating the underlying preferences for each persona to website classifications is another area with future potential.

We conducted many tests, and the contrast among three interim models illustrates the progress we made in tuning the results until we determined the best and final model. This progress shows what's possible when improving NPC realism. The three tests are described below:

1. **The Initial Model—M0 ("Random")**. Our first interim ML model included no NPC preferences, so the results paralleled choosing websites randomly for the NPC to browse. This "random" model is how most browsing patterns are generated today; therefore, this result helps establish a baseline. The average gain for this model is 0%, meaning it generates new browsing lists that are no better or worse than choosing sites randomly.
2. **The Middle Model—M1 ("No Negative Feedback")**. We composed this interim ML model using no negative feedback. Instead, we discarded suspicious matches—removing them from the training data entirely—and supplied only positive match examples. Since the model did not have negative results to balance its choices, our results were worse than choosing sites randomly.
3. **The Final Model—M2 ("Balanced")**. Incorporating what we learned from the previous two interim ML models, we made adjustments—including the number of iterations used, approximation ranks, and so forth—that led to a 26% improvement in an NPC's ability to browse sites that closely match its preferences. We expect to see continued improvement each time we run the M2 model, with the results of one iteration informing the approaches of the next.

We should also note that in each of these models, we did not take the weight of the original preference into account, rather, we simply assumed any preference value over 0 was a positive correlation for that particular preference. In the future, we might use these in comparisons or otherwise when evaluating the model.

See Table 5 for a summary of these findings.

Table 5: Model Testing Results

			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%	1-8%	77%	98%	21%
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%**

7 Future Work

The work outlined in this report is presented as a very simple model for how we might reason about NPCs having particular likes and dislikes, and how these preferences might manifest in their browsing history.

While our results show that an average of an NPC's browsing history is 26% more aligned to its primary preference, we understand that this is a greatly simplified representation of human browsing behavior. There remains great opportunity for future work to expand the notion of personas and the number of preferences that a single NPC might simultaneously maintain.

Similarly, using the results of the model also offers future opportunity to answer questions such as

- Should the length of content an NPC consumes matter? Does long-form content matter more or less?
- Does the frequency of content matter? If an NPC sees content aligned to one preference far more than other preferences, how does that influence the NPC's overall set of preferences?
- If frequency matters, what happens when an NPC saturates a particular preference? Does an NPC switch from its browser to another application to "take a break?"
- How should we reason about negative preferences? What impact do they have for an NPC in relation to correlating positive preferences?
- How do NPCs implement the results of a decision? For example, does the NPC linger on a page longer when it aligns with its preferences?

8 Conclusions

In striving to improve the realism in cybersecurity exercises and training, ML continues to provide a unique opportunity to contribute to (1) creating realistic exercise scenarios and artifacts and (2) processing the resulting realistic data.

In this report, we described one way to use ML to increase the fidelity of training to real-world events and prevent players from exploiting game-isms, which can reduce the exercise's value. We also identified the risk of overfitting ML models. Beyond overfitting, the power and value of ML is too powerful to ignore.

This report summarizes how we improved the realism of browsing within the cyber exercises that we design and execute. First, we created a preference system that enables each NPC to distinguish between multiple choices. For our example, each NPC had a list of preferred websites. We were able to match relevant websites to those preferences using a simple ML approach. For example, an NPC with an interest or preference for computers would browse sites related to that topic more often over time. This behavior is a reasonable approximation of what we might see in real life on a large organization's computer network. This preference behavior is an important part of replicating realistic activity within a cyber exercise at an affordable cost.

References

URLs are valid as of the publication date of this document.

[Chen 2005]
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. Association for Computing Machinery (ACM). 2005. https://doi.org/10.1145/1056808.1056836

[Dobson 2017]
Dobson, Geoffrey B.; Podnar, Thomas G.; Cerini, Adam D.; & Osterritter, Luke J. *R-EACTR: A Framework for Designing Realistic Cyber Warfare Exercises*. CMU/SEI-2017-TR-005. Software Engineering Institute, Carnegie Mellon University. 2017. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=505224

[Hammerstein 2010]
Hammerstein, Josh & May, Christopher. *The CERT Approach to Cybersecurity Workforce Development*. CMU/SEI-2010-TR-045. Software Engineering Institute, Carnegie Mellon University. 2010. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=9697

[Schmidt 1999]
Schmidt, Albrecht; Michael Beigl; & Gellersen, Hans-W. There Is More to Context than Location. *Computers & Graphics*. Number 23. Volume 6. 1999. Pages 893–901.

[Updyke 2018]
Updyke, Dustin D.; Dobson, Geoffrey B.; Podnar, Thomas G.; Osterritter, Luke J.; Earl, Benjamin L.; & Cerini, Adam D. *GHOSTS in the Machine: A Framework for Cyber-Warfare Exercise NPC Simulation*. CMU/SEI-2018-TR-005. Software Engineering Institute, Carnegie Mellon University. 2018. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=534316

[USCC 2019]
U. S. Cyber Command. U.S. Cyber Command Technical Challenge Problems Guidance. USCC-J9-TO-2019-03-12. 2019. https://www.cybercom.mil/Portals/56/Documents/Technical%20Outreach/Technical%20Challenge%20Problems.pdf?ver=2019-07-02-151118-497

[USMC 2019]
United States Marine Corps Training and Education Command; POC: Major Scotty Black, USMC. Intelligent Wargame (i-Wargame): USMC AI MVP Project Submission. 2019.

[Wooldridge 1995]
Wooldridge, Michael & Jennings, Nicholas. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review*. Volume 10. Issue 2. June 1995. Pages 115–152.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	December 2021	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Using Machine Learning to Increase NPC Fidelity	FA8702-15-D-0002

6. AUTHOR(S)

Dustin D. Updyke, Thomas G. Podnar, Geoffrey B. Dobson, John W. Yarger

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2021-TR-005

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
SEI Administrative Agent AFLCMC/AZS 5 Eglin Street Hanscom AFB, MA 01731-2100	n/a

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

Experiences that seem real to players in training and exercise scenarios enhance learning. Improving the fidelity of automated non-player characters (NPCs) can increase the level of realism felt by players. In this report, we describe how we used machine learning (ML) modeling to create decision-making preferences for NPCs. In our research, we test ML solutions and confirm that NPCs can exhibit lifelike computer activity that improves over time.

14. SUBJECT TERMS	15. NUMBER OF PAGES
ML, machine learning, realistic, cybersecurity, training, exercises, scenarios, simulations, GHOSTS, NPC, non-player character	27

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL