Strike Group Defender

G. Mark Jones, Matthew C. Gombolay, Reed E. Jensen, and Steven L. Nelson

Defending U.S. Navy ships from the growing danger presented by modern anti-ship cruise missiles is a formidable challenge. Lincoln Laboratory, partnering with government and industry, developed the game-based trainer *Strike Group Defender* to equip the modern sailor with the knowledge and skills necessary to address the evolving threat. The combination of the immersive interface with novel machine learning and artificial intelligence techniques is advancing the state of the art in interactive training.

From extending the reach of the military, to countering piracy, to defending against ballistic missiles, to responding to natural disasters, the surface fleet of the U.S. Navy performs a range of critical missions to protect national interests at home and abroad. To provide the diverse capabilities required by these missions, the Navy has fielded a fleet of more than 100 major surface combatants, ranging from versatile littoral combat ships and guided-missile cruisers to the immense nuclear-powered aircraft carriers [1]. The total crew size represented by these ships is well in excess of 90,000 sailors, underscoring their importance as a major Navy asset [2].

Though the fleet is an undeniably formidable global force, potential adversaries are developing advanced weapons, intent on putting U.S. ships and their crews at ever-increasing risk [3]. The emerging capabilities and proliferation of modern anti-ship cruise missiles (ASCMs) present a considerable threat to the surface ships of the Navy and their missions [4].

In recognition of this evolving threat, the Navy has a wide array of counter-ASCM systems, both deployed and in development, with the goal of equipping each ship (Figure 1) with a robust layered defense [5]. Each countermeasure system provides unique and complementary capabilities that must be employed quickly, correctly, and judiciously to mitigate the ASCM threat.

While the diversity in defensive systems is designed to enhance robustness for addressing the wide variety of ASCM types, the additional complexity of the combined defensive system presents a significant challenge for the sailors tasked with responding to ASCMs. For example,

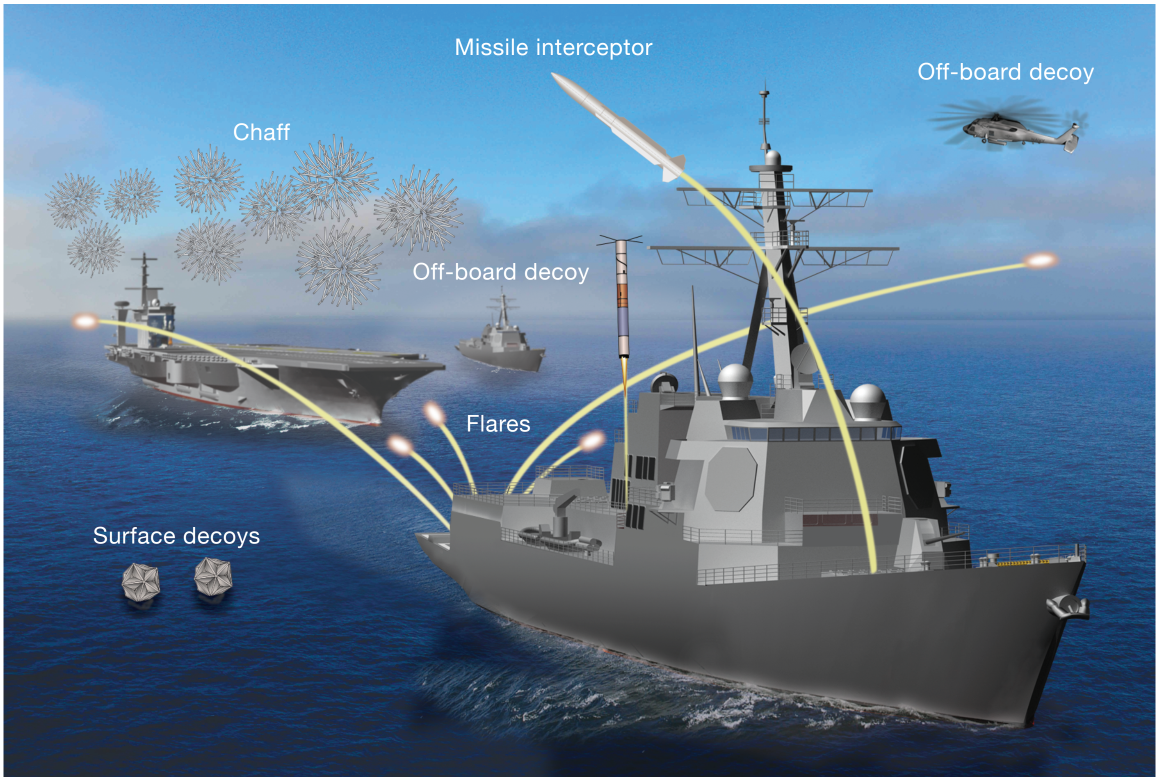


FIGURE 1. The various systems depicted in the illustration can be deployed to counter anti-ship cruise missiles.

assigning five countermeasures against five threats allows more than 100 distinct combinations of countermeasures, though only a few of these potential choices may actually result in a positive outcome for the defense. More realistic situations than this simple example have greater numbers and types of both attacking missiles and countermeasure systems, with additional complications from timing and geometric deployment considerations, and are therefore exponentially more complex. The challenge for the modern sailor is to select the correct course of defensive action, often on very short timelines and with incomplete information, from the large number of choices afforded by the array of countermeasure systems.

A critical factor in preparing the modern sailor to address complex ASCM scenarios is clear, accurate, and detailed training [6]. In recognition of this, Chief of Naval Operations Admiral John Richardson has made one of his four principal thrusts for the Navy to "achieve high-velocity learning at every level." In particular, he suggests that the Navy "expand the use of learning-centered technologies, simulators, online gaming, analytics and other tools as a means to bring in creativity, operational agility, and insight" [7].

The serious game *Strike Group Defender* (*SGD* for short) was designed with this training need in mind, harnessing the immersive nature of modern video game technology, coupled with cutting-edge adaptive machine learning techniques, to provide the Navy with a flexible training and evaluation tool suitable for addressing demanding, realistic modern scenarios. In the end, the goal of *SGD* is to enable sailors to better defend themselves and their ships against the real dangers they face in their naval assignments.

Why a Video Game?

Because the purpose for the vast majority of video games ever produced has undeniably been entertainment, there has been a natural uncertainty and guardedness about games' effectiveness and legitimacy for educational uses [8–10]. Even so, familiar schoolhouse games, such as *The Oregon Trail*, have occupied a niche market in the gaming world since the 1980s [11]. As technology has improved and the proliferation of video games into everyday life has increased, the interest in using video games for education also has grown [12]. The development of *SGD* as a video game was driven by several key factors: the clear connection games provide to young people, the game industry's development of supporting technology, and the natural representation of defense against ASCMs as a two-sided game.

Connection

Part of the growing interest in leveraging video games for instruction flows naturally from the realization that today's students have never known a world without the influence of video games [13]. To put this in perspective for the Navy, where the average enlisted crew member is 22 years old, the original *Mario Bros.*™ game was a quarter-century old when today's sailors were in middle school and the venerable progenitor game *Pong*™ was already approaching 40 years of age [14–16]. Furthermore, the pervasiveness of video games for young people today in the United States can be quantified in part by noting that 97 percent of them report playing some form of video game, whether on gaming consoles, on personal computers, or, increasingly, on mobile devices [17]. The familiarity of the video game medium therefore offers the potential to tap immediately and intuitively into the everyday experience of the target audience of young sailors. The intuitive interfaces and instinctive gameplay developed for *SGD* allow players to focus on learning ship defense rather than on the mechanics of the game itself.

Technology

In tandem with the expansion of the influence of video games, the exponential growth in the computing capability that fuels the industry offers opportunities for developing instructional methods different from those found in more traditional teaching [18]. With educational video games, teachers can take advantage of immersive and engaging on-demand lessons, networked team training, and immediate examination with feedback [19]. In addition, the massive data collection and new analysis techniques supported by a modern video game permit *SGD* developers to explore new avenues for improved and adaptive teaching, training, and testing [20].

Natural Game

While the stakes are extremely high and very real, the defense of a ship against an attack of ASCMs aligns itself very well with the pure definition of a game: two independent sides (the defense and offense) with definite objectives (minimal/maximal damage), operating under certain rules (the capabilities of defensive/offensive systems) [21]. Learning to play the game translates to the core goal of *SGD*: teaching sailors how better to defend their ships in the real world. Additionally, the tense real-time scenarios faced in defense against ASCMs naturally add an element of excitement and entertainment to the game, increasing player engagement and educational opportunities.

The Reality for the Simulation

The game of chess can be intricately complex even though the movements of individual pieces are straightforward to define. In much the same way, the complexity in mounting a defense against ASCMs is derived from the much simpler definition of the offensive and defensive systems that may be employed. Understanding the capabilities provided by these "pieces" is therefore necessary for understanding the nature and magnitude of the complexity found in the overall "game" of ship defense.

The Offense

With significant roots in the technology developed near the end of World War II, the first ASCM was introduced on the world stage in the late 1950s [22]. Since that time, the diversity of ASCM types and their associated array of capabilities have grown steadily, with a world arsenal of more than 75,000 and the number of distinct types in excess of 100 varieties [23].

Though the diversity of ASCM systems is dauntingly large, the number of characteristics needed to define a given system at a high level is comparatively compact. Namely, once the flight profile (how it moves) and terminal seeker (how it sees and thinks) are defined, the system can be modeled sufficiently for the training goals in *SGD*.

Cruise missiles are kinematically diverse, with speeds ranging from subsonic to highly supersonic and altitudes from very high down to low-profile sea-skimming approaches [24]. Additionally, some systems incorporate high-g maneuvers in an attempt to evade missile interceptors fired by the defense [25]. The need for quick decision making by the defense can be brought into focus by considering fast, low-flying threats, for which the time from first appearance above the horizon of the ship until impact can be less than one minute.

Because a ship is a moving target, all cruise missiles have some sort of terminal seeker designed to guide the missile to impact its target. Many ASCMs have radars mounted in their noses for this purpose, but passive sensors (homing in on emissions from the ship) or infrared sensors are also possibilities [26]. While the seeker enables the missile to select among targets and attempt to filter out decoys, it also provides an avenue for the defense to counterattack via electronic warfare techniques [27].

The Defense

Since the advent of the ASCM threat, the Navy has continually developed and deployed a wide range of ASCM countermeasure systems. Though the diversity of systems is large, all of them can be sorted into one of two classes: hard-kill (physically destroying or disabling the threat) or soft-kill (confusing or blinding the ASCM seeker) [28].

The primary hard-kill systems aboard ships are defensive missiles called interceptors, designed to physically destroy the attacking ASCM before it can hit the ship [29]. Much like the ASCM threat, these defensive systems are defined by how far and fast they fly, as well as by the type and capability of their own terminal seekers. Additionally, close-in weapon systems utilizing a high-rate-of-fire gun are also a form of hard kill [30]. Each hard-kill system's strengths and weaknesses determine the likelihood of its effectiveness against a given threat.

In contrast to the dramatic operation of hard-kill systems, the soft-kill systems on the ship employ more subtle means to defeat incoming threats. Onboard and off-board jammers interfere with the operation of the ASCM seeker to blind or confuse its targeting, attempting to render the threat unable to guide to the target ship [31]. Additionally, soft-kill decoy countermeasures can be deployed to act as a more enticing target, causing the threat not to target the actual ship [31]. The performance of soft-kill countermeasures depends heavily on when and where they are deployed and on the capability of the seeker installed on the attacking missile.

The counter-ASCM systems, both hard and soft kill, are supported at some level by the radars on board the ship and off board (e.g., on aircraft), as well as by electronic support measure systems listening for threat seeker emissions [32].

The Game

At the most basic level, the goal for the offense is to inflict as much damage as possible (potentially on high-value ships) with its resources, while the defense seeks to mitigate the damage and conserve its own countermeasures, saving them for potential subsequent attacks.

In the conflict, the offense has some significant advantages, including deciding when the attack will occur, which types of ASCMs will be used, how many will be deployed, how they will be spaced geometrically and in time, and which ships will be targeted. The offense is challenged by two conditions: the target is moving, and the ships in the strike group can operate as a team.

The defense's advantage is that it decides which ships are in the strike group, how they are positioned, and how they are equipped. Challenges for the defense include the uncertain identification of the attacking threats and imperfect knowledge of how many threats will attack at the current time and how many may attack later.

An effective defense requires judicious employment of countermeasure systems, with correct deployment timing and doctrine, in the face of limited information on a very limited timeline. The complexity of this challenge has spurred the continued development of the *SGD* game to help sailors become familiar with the critical decisions they may face and their options.

To maximize the clarity and effectiveness of instruction, a serious game must represent the salient features of the simulated scenario while minimizing extraneous information [33]. *SGD* was designed to provide minimally detailed representations of real offensive and defensive systems while essentially retaining the full complexity of the systems' combined interactions that would be faced by a sailor mounting a defense against the broad array of potential ASCM threats.

Because the task of defending a ship against ASCMs can be seen to fit perfectly in the paradigm of a game, *SGD* was seen as a logical, relevant training tool for the modern sailor.

Genesis of Strike Group Defender

The Navy is pursuing the enhancement of capabilities across a wide variety of new and ongoing hard- and soft-kill efforts. From large programs of record for new radars and electronic warfare systems, to Future Naval Capability efforts, to speed-to-fleet reactions to urgent

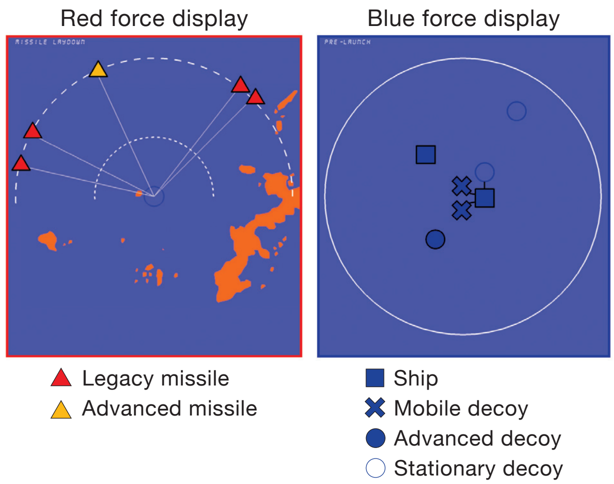


FIGURE 2. The Red/Blue game featured separate displays for the offense (left) and the defense (right).

needs, the breadth of new development is considerable [34–36]. In addition, concurrent with the creation of new systems, new tactics and deployment algorithms are being pursued. The result is that the capabilities and complexities encountered by today's sailors are steadily increasing, and so is the need for education that addresses these technological advances. Lincoln Laboratory's involvement in many of the Navy's development efforts has afforded the Laboratory insight into both the capability and training needs of naval personnel.

The first steps toward the development of an educational video game to address the evolving needs of the Navy were taken during the 2013 edition of the Lincoln Laboratory annual Red/Blue game held by the Air, Missile, and Maritime Defense Technology Division (Figure 2). In contrast to the later expanded *SGD*, the first iteration was focused on conveying several key concepts that illustrate the complexities of defense against ASCMs. This version was constructed as an intense real-time, simultaneous, two-player game, with one player acting as the offense and the other as the defense. The game was publicly introduced during an eight-team tournament conducted in conjunction with the 2013 Air and Missile Defense Technology (AMDT) Workshop at Lincoln Laboratory.

Though the game was comparatively limited in scope at this stage, constructed by engineers (not game developers) and played by engineers (not Navy ensigns), several notable findings emerged from the tournament. First, it was clear that teams that practiced more ahead of time (i.e., trained together) performed markedly better than the less practiced teams. Additionally, teams that identified a limited selection of preferred strategies on both offense and defense were better able to respond quickly to a wide range of their opponent's strategies. And finally, teams that had clear roles identified for each member were able to perform more measured responses, even against unexpected opponent behavior, on a short timeline. Though the observations from the Red/Blue tournament were qualitative, they provided insight into the potential for using a video game construct for experimentation and training.

Based on the initial demonstration of *SGD* during the Red/Blue game at the AMDT Workshop, the Office of Naval Research, PMR-51 Branch, expressed a desire to greatly expand the game into an immersive training and demonstration tool for the Navy. Partnering with professional video game developer Pipeworks and game consulting firm Metateq, Lincoln Laboratory rapidly transformed the Red/Blue game into the first version of *SGD*, which personnel at the Naval Postgraduate School in Monterey, California, beta tested.

Strike Group Defender Functionality

Over a few months, Pipeworks incorporated an array of professional-grade enhancements to expand the concept of the original Red/Blue engineering demonstrator into the first polished iteration of the *SGD* video game, which offered significantly increased training potential. Immediately striking were the improved visuals and stirring soundtrack designed to create an immersive atmosphere and draw the player into the game. The action takes place in a third-person, three-dimensional arena, with the defended ships (blue) in the center and the cruise missile threats (red) flying in from the horizon. Supplementary displays and interfaces are arrayed around the large central display, providing users with easy access to all information needed to play the game (Figure 3).

The player is given complete control over the defense, deciding strategy, deploying countermeasures, and even changing ship speeds and headings. This idealization, along with the representation of the interfaces, is intended to teach core concepts rather than simulating a particular display, piece of hardware, or role for an individual sailor. The supposition underlying the game's design is that sailors more well-versed in the global operations of ship defense will better be able to fulfill their particular roles as part of the crew.

Similar to the idealization of the displays, the missiles and countermeasures in *SGD* are abstract representations intended to convey core concepts rather than represent real systems (Figure 4). On the red side, the types of ASCMs vary primarily with how the missile finds or selects its target. The Moth Missile, for example, uses an infrared seeker to measure heat from ships. On the blue side, the systems are broadly representative of classes of countermeasures. For example, the Hard-Kill system represents the full variety of hard-kill options on a ship.

However, with the understanding that more realism could be desirable for some instructional considerations, the game was designed in such a way that converting to more realistic (and therefore also classified) representations amounts to a straightforward change to the input file defining the system.

FIGURE 3. The display for *Strike Group Defender* gameplay presents blue ships and red threats (center), an overhead view (lower left), a message panel (lower right), a countermeasure inventory (right), and menus and scoreboard (top).

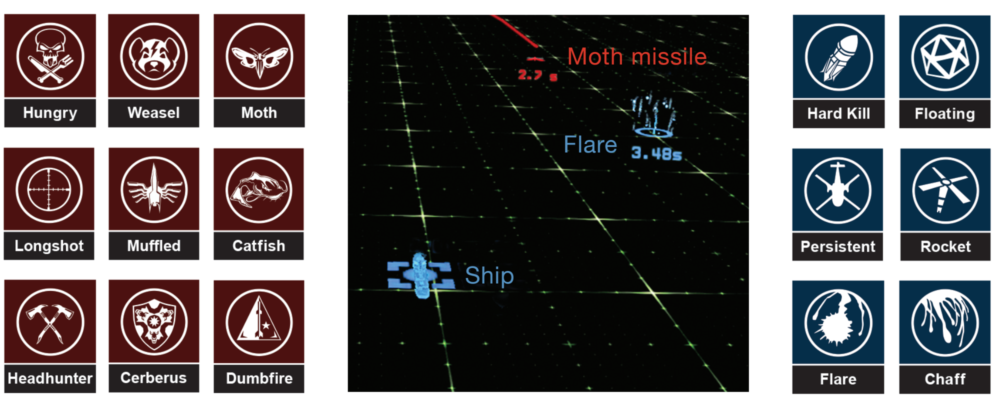


FIGURE 4. *Strike Group Defender* features a variety of abstracted threat missile types (left) and defensive countermeasures (right), each with its own characteristics and capabilities. For example, the Moth Missile can be distracted from the defended ships with a flare (center).

SGD is packaged with a range of built-in scenarios, from one-threat versus one-ship tutorials up to a full strike group versus an attack of 20 missiles or more. The game is not limited to these scenarios, however, as *SGD* also includes a built-in scenario editor that permits instructors and students alike to create their own situations (Figure 5). Ships, countermeasure loadouts (number of countermeasures carried on board), threat type, bearing, and timing are all adjustable, allowing players to explore actions from the point of view of both sides of the conflict. Additionally, threat timing and bearing can be varied, even randomly, adding challenge and discouraging rote memorization of responses.

Scenarios are played in real time, typically last a few minutes, and are playable under a variety of game modes that provide different instructional opportunities for the user:

- Tutorial. Straightforward scenarios with a single type of incoming threat coupled with a virtual instructor teach players where and when to deploy the correct countermeasures.
- Single-player defense. Controlling a single ship or group of ships, users defend against a computer-controlled ASCM attack in a variety of scenarios across a range of difficulty levels based on the number of incoming threats and the availability of countermeasure resources.
- Multiplayer defense. Through in-game text messaging or over a voice network, multiple players collaborate in real time to defend surface ships against a computer-controlled ASCM attack.
- Multiplayer offense versus defense. One player controls the adversary ASCMs (i.e., offense) while all other players collaborate as the defense. This setup enables players to gain insight into potential adversary strategies and the tactics to counter them.

In addition to the central gameplay functionality, *SGD* also incorporates many social features designed to increase player interaction, encourage competition, and ultimately improve learning (Figure 6). Each scenario has its own leaderboard on which top scores are continuously updated for all players to see. The innate desire to be atop the leaderboard is a powerful motivating force for individual improvement [37]. Similarly, the ability to create and share new scenarios with which to challenge other players is also intended to foster creativity and the desire to improve.

FIGURE 5. The editor panel allows the player to construct a scenario, selecting which type of threats to confront, how many, and where and when they are deployed against the elected defended ships.

FIGURE 6. The intuitive interface offers players easy access to scenarios, leaderboards, and social media.

Finally, the message board facilitates communication among the players, allowing them to ask questions of their peers and instructors and to share insights gained.

Though *SGD*'s capabilities are extensive, the game was designed from the beginning to require minimal system requirements to work properly. Running in any web browser with very low bandwidth requirements, the game retains full functionality whether played on a desktop computer in the classroom, on a laptop at home, or over secure networks on ships deployed at sea.

SGD was introduced to the wider community at the 2014 Air and Missile Defense Technology Workshop at Lincoln Laboratory. Over the three days of the workshop, 67 participants logged 332 games. The positive feedback from the community qualitatively validated many of the underlying motivations that had influenced the development of *SGD* and reinforced the Navy's desire for the research team to pursue further enhancements.

Strike Group Defender Roles

The diverse capabilities designed into *SGD* enable access to multiple training avenues with the ultimate goal of equipping sailors with the skills needed to defend their ships within complex threat scenarios. These educational opportunities can be broken down into three distinct categories: teaching, exploration, and evaluation.

Teaching

The capabilities of *SGD* enable rapid instruction in ASCM defense, from threat characteristics, to countermeasure capabilities, to implementation of correct tactics. Using the built-in capabilities, instructors can construct lesson plans to relate core concepts in the classroom setting, or students can experiment on their own.

One of the notable benefits of *SGD* is building sailors' trust in new capabilities. In response to the continually evolving ASCM threat, the Navy is rapidly introducing new countermeasure systems to the fleet. In particular, the new soft-kill systems, composed of a variety of onboard and off-board jammers, may seem arcane and untrustworthy if one does not understand how they actually do the job of defeating ASCMs. Because the new systems are unfamiliar to sailors, they may have a tendency to downplay these systems in favor of older, more familiar ones. By observing in *SGD* how new systems operate, sailors can learn how the systems work and therefore may choose to employ them appropriately in the field.

Exploration

In contrast to the cost of making a mistake in countering a real ASCM, the penalty for performing poorly in *SGD* is only a lower game score and the immediate opportunity to try to improve. This lack of consequences encourages players to experiment and to try any "what if?" scenarios desired. In this way, a deeper understanding of core concepts can be attained, and new methodologies may even be discovered [38]. Because the feedback is immediate, the trainee can try a wide variety of approaches in a short amount of time.

The game also permits outside input, which could, for example, come from another computer executing a new algorithm designed to help sailors do the job of ship defense. Thus, *SGD* can serve as a proving ground for new technologies with which sailors can interact to improve ship defense.

Evaluation

The construct of a video game, in which everything can be quantified, can provide educators with many opportunities for evaluating students' success at the tasks of the game. The *SGD* environment records a large amount of information, ranging from the number of missiles correctly mitigated, to the number of resources expended, to reaction time, to deviation from desired tactics. The availability of these data affords instructors wide latitude in evaluating the performance of *SGD* users.

Emergence of Machine Learning

As the ASCM threat has grown in numbers and complexity, so too must the capabilities of naval training grow. The fusion of a video game interface, massive data collection, and modern machine learning techniques presents a potentially powerful and nontraditional mode for enriching training for the sailors of today and the future.

The behind-the-scenes data collection built into *SGD* is no less important than the eye-catching graphics and intuitive interfaces of the game. Every action of every user in every game is seamlessly recorded into a massive data archive that allows every game to be replayed and studied by any player. This replay functionality has the benefit of allowing trainees to learn from their own successes and mistakes, and from those of other players. Beyond that, these collected data enable the game to "learn about" its players and adapt itself to their needs. Through this application of cutting-edge machine learning techniques to the *SGD* data, the instructional capability of the game is maximized, and each user is ensured an experience tailored to his or her particular learning style.

Tournament Data Collection

To demonstrate the utility of applying machine learning techniques in *SGD*, a large dataset for experimentation was needed. To fill this need, a Laboratory-wide *SGD* tournament, designated March Madness, was conducted. Beyond the intrinsic draw of competition, the Lincoln Laboratory Director's Office further encouraged participation by offering a cash prize to the champion.

Because *SGD* requires only modest computing power and functions on most any platform, the tournament could be played on demand over the local network on regular desktop computers. In the initial competition round, players attempted a variety of challenging scenarios. The top 16

What is Machine Learning?

Machine learning is a subfield of artificial intelligence in which researchers develop computational methods (i.e., algorithms) that give computers the ability to autonomously learn a model to explain data. Generally, machine learning is categorized into two branches: supervised and unsupervised.

In supervised learning, the goal is to predict an outcome from a previous example. For example, suppose a meteorologist who wants to predict whether it will rain tomorrow has data from over the previous 50 years that tells, for each day, the temperature, humidity, barometric pressure, and wind speed. These data are known as the features and are represented as a numeric vector, denoted $\vec{x}$, which describes each day. For each day, the meteorologist knows whether it rained the next day. This datum is known as the label, denoted y, for the corresponding features. The goal is then to learn a mapping $f: \vec{x} \rightarrow y$ to predict whether on any given day, described by $\vec{x}$, whether it will rain, y. This example is a classification task: the prediction variable can take on one of a finite number of values. In this case, the outcome is binary—either 0 or 1 describing whether or not it will rain. Conversely, a regression task involves predicting a continuous value, such as tomorrow's high temperature. Other examples of supervised learning include predicting whether a camera's image contains a person of interest (i.e., facial recognition), translating Arabic to English, or determining the correct medical diagnosis for a sick patient. Common techniques for supervised learning include logistic regression, decision trees, support vector machines, neural networks, and *k*-nearest-neighbors.

The complement of supervised learning is unsupervised learning. The goal of unsupervised learning is to infer a function to describe one or more hidden attributes within data. Consider an example from U.S. politics, specifically, the legislative branch. Let us say we knew that congressional representatives voted yea or nay on certain bills, and we had features describing those bills. Rather than predicting whether a representative would vote yea or nay on a future bill as in supervised learning, now we want to group representatives according to similarity. The hidden attribute is party affiliation. If we give the unsupervised learning algorithm the features of bills along with how each representative voted, the algorithm will output an assignment of each representative to a group. If this algorithm was able to mimic reality, it would learn there are three groups: Republicans, Democrats, and Independents, and it would assign each representative to one of those groups. The key is that the unsupervised learning algorithm does not know beforehand the notion of a Republican, Democrat, or Independent—just that there are groupings of some kind. In addition to our political example, other unsupervised learning tasks might involve learning taxonomy for life on Earth (i.e., how to group life by species, genus, family, etc.) or learning a grouping for people according to the types of movies they watch. Common techniques for unsupervised learning include *k*-means, Gaussian mixture models, self-organizing maps (a type of neural network), and principal component analysis.

players then competed in a single-elimination tournament; ultimately, the champion of a final-four series was crowned at an event complete with an audience and announcers.

Participation in the tournament exceeded expectations, with 140 players completing nearly 3,000 games, totaling approximately 100 hours of game time. The draw was diverse, with many players having no direct experience with the Navy in general or ship defense in particular. One player, a graphic artist who made the Sweet 16 as a top player, notably commented, "I have been illustrating these concepts for years, but now I understand what they mean." Beyond the data collected, this sort of anecdotal evidence about the benefit of *SGD* helps validate the game's educational value.

From Analysis to Enhanced Education

As players learn from *SGD*, the game is also learning from them. While the immense amount of data collected by the game holds the promise of increasing the effectiveness of instruction, it also presents an immense challenge to the analyst to distill the data down into a meaningful and useful result. Machine learning techniques are ideally suited to this situation, revealing hidden correlations and providing instructive adaptability useful to both trainees and teachers.

An exploration of the utility of these techniques began with a multidimensional analysis of the data collected from the internal Lincoln Laboratory tournament [39]. Presented here is a subset of the results of that analysis, categorized into three topics: identifying player types, identifying tactics, and adaptive lesson planning. Each of these topics has immediate relevance to meeting the Navy's educational needs for addressing complex ASCM scenarios.

Identifying Player Types

Players approach video games with a range of styles and motivations, and, similarly, there is diversity in learning approaches [40, 41]. Categorizing players based on the features measured by *SGD* is a critical first step to enable instruction that adapts to the natural tendencies of each trainee. Moreover, the characteristics of high-performing game players can be reinforced, while the approaches of lower-performing players can be identified and remedied with tailored instruction.

Critical to player typing is the collection of large amounts of data, made possible through *SGD*'s harnessing the continuous monitoring and recording of player actions enabled by modern video game technology. Which countermeasures players use and when, how quickly they react to changing situations, and which tutorials they attempted and completed and in which order, all can be used as features to help define each player.

The large number of data points and the high degree of dimensionality provided by the individual features measured by *SGD* can be reduced to a manageable set of categories through the application of unsupervised learning (i.e., clustering) techniques. The characteristics that define the players, beyond just a score or a letter grade, are then brought into focus. Players are not only categorized, but the deeper explanations for their performance can begin to be explored.

As a qualitative example of clustering, consider the classification of automobiles. The diversity in make, model, and model year is quite large, analogous to the number of players of *SGD*. Similarly, the number of features used to define a car could also be large, such as cost, performance, fuel efficiency, reliability, safety, and cargo space. Clustering could be used to determine a more concise set of automobile categories (e.g., family, utility, sport, or luxury) and the associated features that define each category. The complexity of the data is thereby reduced to a more manageable and useful level of categorization.

A more quantitative two-dimensional example of data clustering is depicted in Figure 7. Each point in the dataset is represented by two features: its x and y coordinates. The simulated input dataset is color-coded in Figure 7a to show that it was generated from five overlapping sources. In Figure 7b, the color-coding has been removed, illustrating that the underlying structure is not apparent. The challenge for effective clustering is then, given Figure 7b as an input, to extract some approximation of Figure 7a as an output.

One way to infer the true clusters underlying a dataset is to apply the k-means algorithm [42]. The process supposes a number of clusters (represented by the k in its name) and then attempts to partition the data, minimizing the cumulative distance metric seen in Equation (1).

$$\text{Cumulative distance} = \sum_{j=1}^{k} \sum_{x_i \in s_j} \left\| x_i - \mu_j \right\| \qquad (1)$$

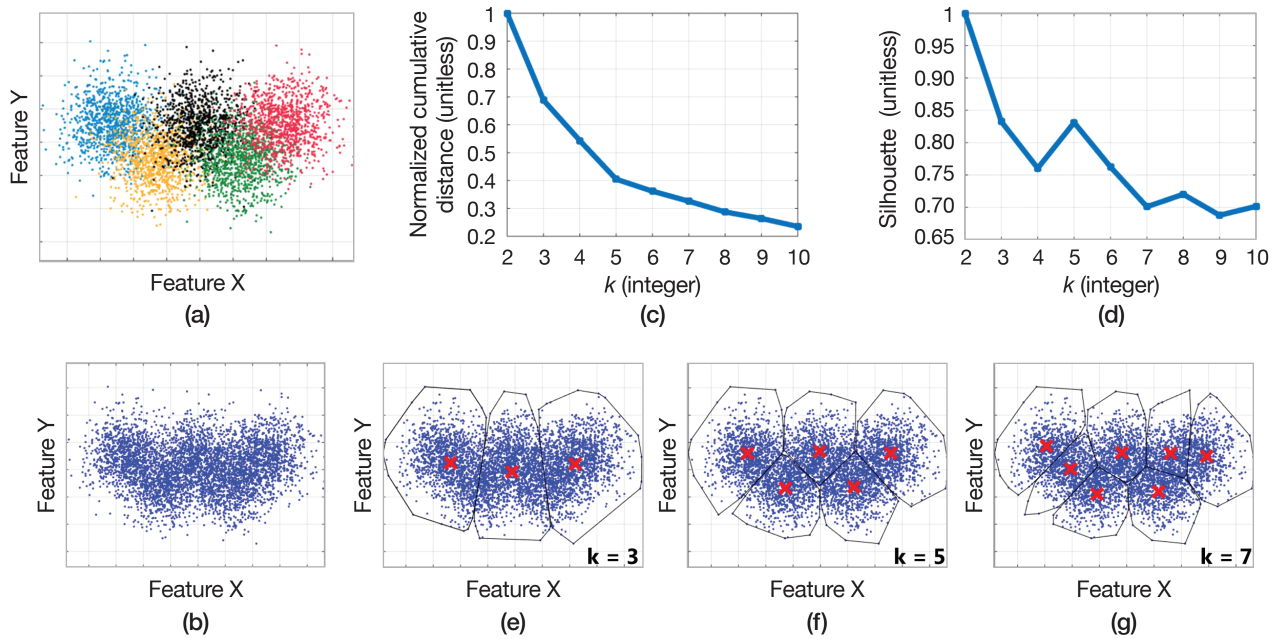


FIGURE 7. The plots show the results of *k*-means data clustering, where *k* represents the number of clusters. The dataset was generated from five overlapping sources, color-coded in (a) and with the color-coding removed in (b). In (c) and (d), the (correct) value *k* = 5 is seen to balance the goal of simultaneously achieving low cumulative distance and high silhouette (i.e., similarity of cluster members to each other). The clustering result for *k* = 5 is shown in (f), and less well-matched fits of *k* = 3 (e) and *k* = 7 (g) are shown for reference. Clusters are outlined in black, and red X's indicate the clusters' centroids.

Clustering minimizes the cumulative distance over k clusters, defined as the total distance of each data point x_i in cluster S_j from the associated centroid μ_j.

The primary output of the algorithm is the centroid of each identified data cluster, defined as the average of the features of all the points in the cluster. Each data point is closer to the centroid for its associated cluster than to any other centroid. The cumulative distance, defined as the total distance summed over all data points to their respective centers, is minimized.

The more clusters (higher k) assumed by the algorithm, the smaller the cumulative distance will be since all points will necessarily be closer to their assigned cluster centroids. Reducing the cumulative distance is a good thing up to a point, but if too many clusters are added, the whole purpose of dividing the data into more manageable partitions is lost.

Therefore, the desire to increase the number of clusters (k) is balanced against the separability power of the fit. One particular metric, known as the silhouette (Equation [2]), allows us to quantify this feature [43]. Maximizing this metric ensures that the distance from each point to the second-closest centroid is maximized.

$$\text{Silhouette} = \sum_{i=1}^{N_{points}} \frac{b_i - a_i}{\max(a_i, b_i)} \qquad (2)$$

The silhouette metric compares the cumulative distance of each data point to all other points in its associated cluster a_i to the cumulative distance to all other points in the second-closest cluster b_i.

Effective data clustering then seeks simultaneously to minimize the cumulative distance metric (Equation [1]) while maximizing the silhouette metric (Equation [2]). In Figures 7c and 7d, the value k = 5 can be seen to indeed best satisfy these criteria. The clustering result is shown in Figure 7f, with less well-matched fits of k = 3 (Figure 7e) and k = 7 (Figure 7g) shown for reference. The clusters are identified by the black outlines, with their respective centroids depicted with a red X.

For the *SGD* tournament dataset, the following set of features was identified as most relevant for use in player typing:

1. Quit rate. Fraction of games that the player quit before the end of the scenario
2. Unique tutorials. Number of unique tutorials attempted by the player
3. Tutorial rate. Number of tutorials attempted by the player
4. Test rate. Number of times the player attempted a test level
5. Tutorials per test. Ratio of tutorial levels to test levels attempted by the player
6. Repeat rate. Number of times the player replayed a level already completed
7. Pause time. Average amount of time the player paused the game (when allowed)
8. Tutorial repeats. Mean number of times players attempted tutorial levels

Features identified with rate were normalized to the total number of games the player had played. Both the number of features (eight) and the amount of data (100 hours of gameplay) are quite large, making the exact solution of Equation (1) impractical computationally. An expectation maximization algorithm was therefore employed to allow a rapid approximation of k-means clustering to be applied to the data [44].

The result of clustering with these features on the *SGD* tournament data was the identification of four player types, shown in Figure 8. The first player type is notable for a significantly higher score in the *SGD* tournament, compared to the other three types. Paradoxically, the first player type is also distinguished by feature 1, a high rate of quitting scenarios. On the surface, this behavior would seem to be a bad quality for a player to exhibit. However, when paired with the high scenario-repeat rate also shown by this group, a play style begins to emerge: when players in this group discern that a scenario is going poorly, they quit and begin anew, immediately attempting to correct their mistake.

The other three groups performed similarly in the *SGD* tournament, though their play styles were different. The second and fourth player types both played a high rate of tutorials, differentiated mainly by the second group opting to quit scenarios while the fourth group used the pause feature more often. The third group eschewed almost all training and jumped right into the

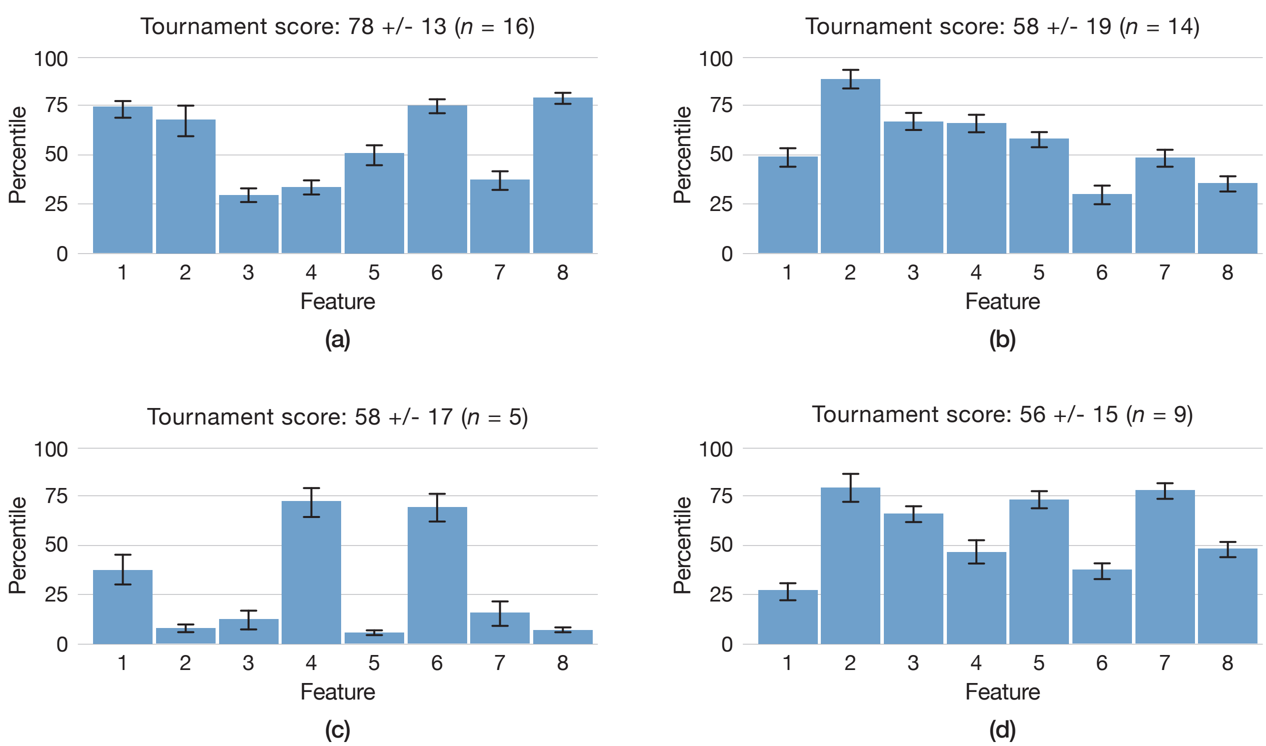


FIGURE 8. Four player types, defined by eight unique features, were found by clustering the data from the *Strike Group Defender* tournament. The first type (a) also corresponded to the highest-scoring players. The remaining types (b, c, and d) scored similarly but had very different approaches to the game, as seen in their feature profiles.

tournament, choosing to optimize their performance only on the examination levels.

Player typing gives a window into how players approach the game and which strategies produce the more desirable outcomes. For example, if group 1 is identified as the preferred way for players to perform, lesson plans could be developed to foster the characteristics of this group in all players. Hand-in-hand with this approach, players quickly can be assigned a type as they play the game, allowing early intervention to either encourage their current approach or to correct unwanted characteristics. Through rapid player typing, the opportunities to improve the performance, and thus the training, of *SGD* users are increased. For instructors in diverse educational settings, similar player typing could inform the development of lesson plans that use video games.

Identifying Tactics

Just as clustering can be used to distill player behaviors down to a few manageable categories, it also can be applied to discover the general classes of tactics employed by players for a given scenario. The results can be used in a traditional educational sense, with instructors confirming that the trainees are indeed employing the tactics that they have been taught. Additionally, the process also allows information to flow the other way: the game can learn interesting nonstandard tactics from the players. The large number of players, combined with the freedom afforded in the gameplay of *SGD*, allows the potential for the creation of enhancements to standard tactical approaches. Thus, identifying player tactics enables improvement of both the trainees and the educational information itself.

In the application of clustering algorithms to the identification of player tactics, the features to be considered present additional complexities: time (when an action is taken) and space (the bearing of the countermeasure deployment) are integral to defining the basis feature set.

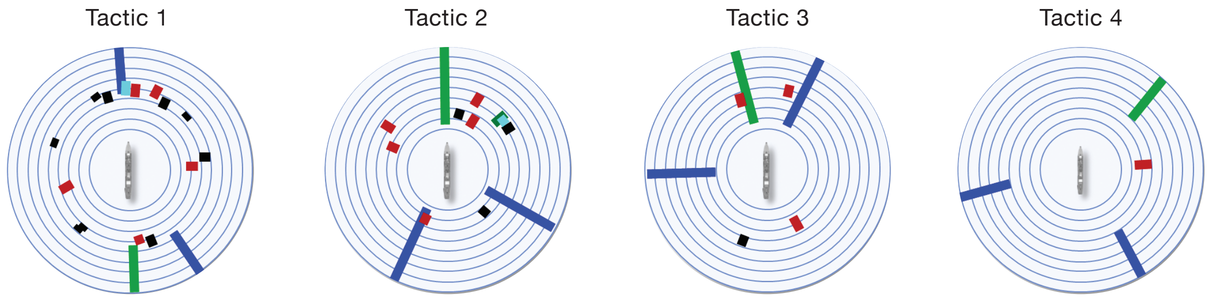
To cluster tactics, the k-medoids approach is used [45]. In contrast to k-means, where a continuum of potential centroid positions is possible for each cluster, k-medoids requires that cluster centroids be positioned precisely on an actual tactic that was employed in a particular game played. This distinction is made because it does not make sense to average individual games played to produce a “mean tactic.” Put another way, deploying a countermeasure successfully to the left in one game and successfully to the right in another game does not imply that deploying it straight ahead is a viable tactic.

Like k-means, the k-medoids algorithm also seeks to minimize a cumulative distance function, as in Equation (1). However, here we are using disparate features that are difficult to compare directly. For example, deploying a rocket-type or persistent countermeasure may be seen as similar tactics, while deploying a flare would necessarily be regarded as different. To account for the variety in actions that may be taken by a player, a weighting scheme was constructed to define the comparisons among all the features making up each game [46]. With this machinery in place, the k-medoids algorithm can be applied to produce clustering results for player tactics.

To provide adequate data for clustering, participants in the *SGD* tournament were encouraged to play the Daily Performance Evaluation scenario, in which threat types were randomized for each game. The tactics extracted from the Daily Performance Evaluation data were found to cluster into four groups (not necessarily corresponding to the four player-type clusters). The prototypical tactic for each group is shown in Figure 9.

The rings around the overhead depiction of a ship represent time in the scenario, with the start time at the innermost ring and the end of the scenario occurring at the outermost ring. The colored lines indicate which countermeasure type was deployed, and on which bearing. While the tournament scores are similar, the tactics are ordered with increasingly successful performance from left to right.

The rightmost tactic came to be known as the Iron Triangle, independently identified by those that played the game. Here, the long-lived countermeasures, such as persistent or floating decoys, are deployed in a triangle around the defended ship to address a range of threats, with the player left to focus on deploying expendable countermeasures as needed to address threats not otherwise defeated. The middle two techniques are variants on this theme, with a few more countermeasures used in the second one and with the geometry a little off in the third. In contrast to the other more measured approaches, tactic number 1 is more sporadic. Recognized as an inefficient “kitchen-sink” approach, large numbers of all countermeasure types are applied continuously against the threats.



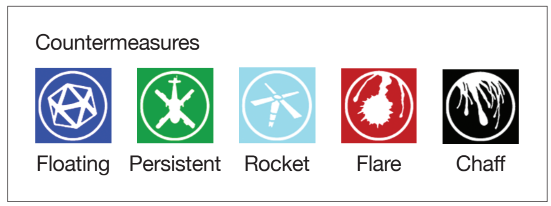


FIGURE 9. The representative player tactics derived from *k*-medoids clustering are illustrated by the four circles. Each concentric ring indicates a time in the scenario, and the color-coded lines indicate the type of countermeasure deployed. From left to right, the tactics are ordered according to increasing effectiveness.

In the context of the scenario analyzed here, the identification of tactics allows for adaptive instruction to encourage players who are already using tactic 4, to prompt players to tweak their tactics if they are using approach 2 or 3, or to teach players to totally overhaul their approach if they are using tactic 1. For other, even more complex scenarios, it is possible that tactics not previously identified as favorable could emerge, helping the game learn the preferred tactic from the players themselves.

Adaptive Lesson Plan

The continuous collection of data by *SGD* enables instructional adaptation in response to the changing needs of each player. Essentially, the game can learn how its players learn and use that information to improve its own teaching. Much as real teachers tailor their instruction to meet student needs, so too can the game adapt its interactions with the players to improve its instructional effectiveness.

As an illustration of the utility of the adaptive teaching concept, the *SGD* tournament data were analyzed to construct an adaptive lesson planner, one that could guide players through tutorials and tests on a path to maximize learning. Through contrasting the learning approaches of the lower- and higher-performing players, preferred approaches were identified. Ultimately, this approach is intended to enable the creation of an on-demand, personalized virtual instructor, one that can observe if a student is headed down the right path and give reinforcement or give correction if the student has gone astray. The potential for instruction tailored to individual students is of considerable interest to the education community [47].

While human learning is a very complex process, significant progress toward a viable virtual instructor can be made with a tractable simplified model of a person's learning [48]. To that end, a hidden Markov model (HMM) was applied to the data collected in the *SGD* tournament [49]. In this type of model, observable states, with transitions between them, are mediated by unobserved states, hence the "hidden" in the name. The model seeks to quantify transition probabilities among the states, allowing for evolution of the system to be predicted.

In the context of the model applied to *SGD*, the observable states are identified as the various tutorial and game levels available to the players. One can train an HMM on particular player types and, because the HMM is generative, create an ordered list of likely game levels. By training the model on high-performing players, game developers can create a positive lesson plan (i.e., a sequence of lessons). Similarly, by training on lower-performing players, a poor lesson plan can be produced. Players who are seen to naturally be on a positive plan can be encouraged while those on a less optimal plan can be redirected.

The HMM topology applied to the *SGD* data is depicted in Figure 10. The observable states, the tutorials, and game scenarios are depicted in the boxes at the bottom. The hidden states, which imply the players' unobservable inner machinations, are represented by the three circles at the top, designated X_1, X_2, and X_3.

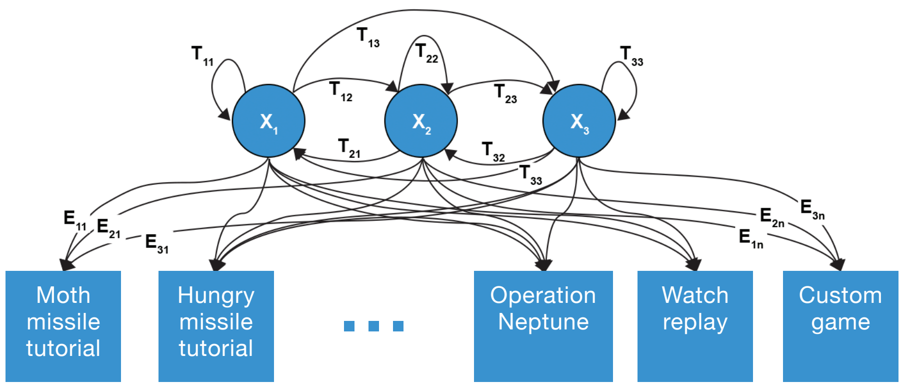


FIGURE 10. This hidden Markov model was employed to quantify the way players traverse *Strike Group Defender*, moving among the different scenarios and games represented by the lower boxes.

In between the circles are transition probabilities, represented by the T_{xy} lines. The transition probability to an observed state, typically known as an emission, is represented by an E_{xy} line. The values for all transitions and emissions are obtained by training the model on the data collected by *SGD*.

To create the adaptive lesson plan model from the *SGD* tournament data, the model was trained on data from two groups of players: the upper and lower 50 percent of performers, identified by tournament scores. The result is two complete hidden Markov models, one demonstrating how the higher-performing students navigate through the game levels and a corresponding model for the lower-performing students. In both cases, the models can then be used to recommend the next level for a student to attempt, given the level just completed. Two example lesson plans so generated from the models are shown in Table 1.

The poor lesson plan on the left side of the table shows players bouncing around between lower-level tutorials, likely making little progress. Players following the better lesson plan, on the right side of the table, appear quickly to jump into difficult challenges. It is possible that these generated lesson plans are merely indicators of player capability and may not directly stimulate player improvement. However, armed with this knowledge, the game itself can attempt to steer players onto an assumed positive path through suggestions about which levels to attempt next and evaluate player improvement along the way.

Though our initial lesson plans are derived from a simple model trained on limited data, they give an indication of the educational advantage that could be achieved with an adaptive instructor built into a game. Future enhancement may include injecting a modicum of recursion into the Markov model to better include effects of a player's history as he or she traverses the game. The true impact of this approach will be quantifiable through the collection of more data and measurement of the performance change in players provided with the adaptive tool.

Table 1. Lesson Plans Depicting the Actions of Two Groups of *SGD* Players

LESSON PLAN GENERATED WITH DATA FROM BOTTOM 50% OF PLAYERS	LESSON PLAN GENERATED WITH DATA FROM TOP 50% OF PLAYERS
Basics Tutorial	Basics Tutorial
Missile Type 1 Tutorial	Challenge Mission
Basics Tutorial	Test
Missile Type 1 Tutorial	Challenge Mission
Missile Type 1 Tutorial	Challenge Mission

Automating Players through Apprenticeship Scheduling

While the previous learning applications apply static analysis to improve a user's experience, imagine if one could dynamically adapt content in real time to a specific player's needs. Recently, Gombolay et al. have pioneered a method called apprenticeship scheduling that learns how to mimic scheduling tasks from expert scheduling demonstrations [50]. In *SGD*, Gombolay et al. showed that the tactical weapon assignments made by a player correspond to a multi-agent, multi-task, time-extended scheduling problem with complex dependencies, one of the most difficult scheduling categories. Using *SGD* tournament data, they were able to learn and mimic individual player behaviors autonomously within *SGD* via apprenticeship scheduling.

Having a learned scheduler opens the door for several real-time user interactions. For example, during an intense battle or a period of information overload, a player could be given prompts to deploy weapons in an expected way, as he or she would typically use them. Using a prompt framework, *SGD* can measure player responses to these suggestions and enable future studies of algorithm trust, acceptance, and reliance. An apprentice scheduler also enables the training of real-time autonomous agents that could either exploit the player's weaknesses or cooperate by offering prompts or filling in actions that the player might often neglect. Players could iterate with the automated adversary or a teammate to learn and improve upon weaknesses or to form a trusted, dynamic team. Having dynamic learning and feedback in *SGD* enables important studies on autonomy, human-machine interactions, teaming, and trust.

Next Steps

Machine learning techniques have a voracious appetite for data, and the studies undertaken with *SGD* are no different. As more people play the game, the dataset for analysis will grow, and the models based on it will become correspondingly better. Additionally, more data will lead to a more quantifiable assessment of the true benefits of the education tools provided by the game.

To date, the data used to explore machine learning concepts have been based primarily on the *SGD* tournament dataset. While much progress has been made, these data were collected on Lincoln Laboratory employees rather than on the true final audience, the sailors in the fleet. Expansion into this area is being facilitated by the Naval Postgraduate School, which has made *SGD* available for play by anyone in the military. The data collected from this forum can be analyzed in the same way as those from the Laboratory's tournament, and it will be illuminating to compare and contrast the extracted results.

In recent months, several improvements have been made to the *SGD* back-end to support interactions with external simulations and artificial intelligence (AI). An application programming interface (API) has been designed to accommodate external models, simulations, and decisions. Enhancements with the API include the ability to send customized prompts to a player and the ability to control the *SGD* simulation time step. Efforts are under way to reduce the simulation runtime to enable AI routines that rely on running many *SGD* instances in order to make a decision. All of these improvements can be combined with machine learning concepts to create a dynamic, adaptive learning environment not available in the Navy today.

While the back-end development of machine learning techniques and AI has been ongoing, the front-end video game has also undergone considerable development (Figure 11). The tactical focus of the first version of *SGD* has been expanded dramatically to include scenarios that require pre-attack strategizing. Full missions take place on a world map. Intelligence, surveillance, and reconnaissance (ISR) resources are built into this updated version, and new scenarios challenge players to avoid the threat of ASCMs in the first place. However, if missiles are launched in the game, the players are drawn into the original tactical-view version of *SGD*, attempting to defend their ships.

Also under development is a classified version of the game that allows for more realistic scenarios to be represented. With new scenarios and mission contexts, new weapon and sensor capabilities can be prototyped and assessed at a high level. In future versions of *SGD*, a player could configure a ship loadout or "purchase" a new weapon or AI capability and determine how well it supports the mission. Recorded player choices could also be used offline to seed an algorithm that solves for optimal loadouts and configurations. With the incorporation of these enhancements, *SGD* is envisioned as transforming from a pure focus on ASCM defense to a broader learning and technology development ecosystem that will enable the exploration of a wide variety of issues for the Navy.

Future Directions

The current research into the benefits of machine learning paired with the *SGD* platform provide a window into the training envisioned for the future. Identification of player types, for example, will help the Navy identify skilled players and also indicate ways to improve the performance of lesser-skilled players. Similarly, the identification of tactics will help identify which responses are effective, with real potential to also harness the creativity of sailors and to learn from them. The adaptive lesson plan personalizes

FIGURE 11. The next iteration of *Strike Group Defender* adds a strategic layer to the original game, with players maneuvering on a world map, attempting to avoid anti-ship cruise missile confrontations.

the learning experience for each player, offering a path to more efficient and focused instruction. The incorporation of autonomy and apprenticeship scheduling enables real-time, adaptive learning that can be tailored to players' needs. These concepts, coupled with a host of other machine learning–enabled approaches, represent a new level of training customization and engagement.

Over a short time period, the original internal Lincoln Laboratory Red/Blue game has been developed into *Strike Group Defender*, a professional video game that is coupled to back-end data storage, extended by an external API, and enhanced by AI and machine learning techniques. This combination is opening up new avenues of instruction and the ability to quantify effectiveness through the analysis of very large sets of collected data. The ultimate goal is to be able to say confidently that we have equipped the sailors in harm's way with the knowledge and skills necessary to address the threats found in challenging modern scenarios.

Awards and Recognitions

The *SGD*'s professional video game development, founded in sound technical concepts and coupled with machine learning technology, has led to recognition for the game by several government and commercial entities. In 2014, the game was recognized as the Best Government Game at the Serious Games Challenge and Showcase at the national Interservice/Industry Training, Simulation and Education Conference. The following year, the National Training and Simulation Association honored *SGD* with the team award for best training game. Finally, the MOVES (Modeling, Virtual Environments and Simulation) Institute at the Naval Postgraduate School in Monterey, California, has said in an assessment of *SGD*: "We recommend the Navy take advantage of this advancement in technology and training consistent with the recommendations being developed and put forward by the Navy Warfare Development Command (Chief of Naval Operations designated lead for Electromagnetic Maneuver Warfare)" [51].

Acknowledgments

The development of *SGD* was sponsored by the Office of Naval Research, PMR-51, with Scott Orosz guiding the work throughout, ensuring that *SGD* was developed into a tool that would meet important needs for the Navy. Russel Phelps of Metateq was instrumental in guiding the team, enabling the engineers at Lincoln Laboratory to interface seamlessly with the developers at Pipeworks and the sponsor. The team at Pipeworks, including Lindsay Gupton, Forest Ingram, and Simon Strange, quickly and expertly changed the modest Red/Blue game into a challenging, useful, and even entertaining serious game. Perry McDowell of the Naval Postgraduate School provided useful insight, ensuring *SGD* was relevant for the real needs of Navy instructors. Without the contributions from these and others, *SGD* would not be on the frontier of education for the Navy today. ■

References

1. Congress of the United States, Congressional Budget Office, "An Analysis of the Navy's Fiscal Year 2016 Shipbuilding Plan," Oct. 2015, p. 3.
2. S. Saunders, "United States," *Jane's Fighting Ships*, Alexandria, Va.: IHS Global Limited, 7 Aug. 2015, pp. 921–960.
3. A. Feickert, "Cruise Missile Proliferation," Congressional Research Service Report for Congress RS21252, 28 July 2005.
4. D.M. Gormley, *Missile Contagion: Cruise Missile Proliferation and the Threat to International Security*, Appendix A, pp. 177–180. Westport, Conn.: Praeger Security International, 2008.
5. M. Meyers, "Future Naval Capabilities (FNC) Program Overview," briefing given at Naval Future Force Science and Technology Expo, Washington, D.C., 4–5 Feb. 2015.
6. Naval Education and Training Command, "Strategic Plan 2013–2023," 2013, p.6.
7. Admiral J.M. Richardson, Chief of Naval Operations, "A Design for Maintaining Maritime Superiority," Version 1.0, U.S. Navy release, 5 Jan. 2016, p. 7.
8. S.S. Adkins, "The 2016–2021 Global Game-Based Learning Market," Serious Play Conference, University of North Carolina, Chapel Hill, N.C., 26 July 2016, p.7.
9. T. DiChristopher, "Digital Gaming Sales Hit Record $61 Billion in 2015," CNBC, 26 January 2016.
10. J.W. Rice, "New Media Resistance: Barriers to Implementation of Computer Video Games in the Classroom," *Journal of Educational Multimedia and Hypermedia*, vol. 16, no. 3, 2007, pp. 249–261.
11. J. Lussenhop, "Oregon Trail: How Three Minnesotans Forged Its Path," City Pages, 19 January 2011.
12. A. Gershenfeld, "Mind Games," *Scientific American*, vol. 310, 2014, pp. 54–59.
13. L.A. Annetta, "Video Games in Education: Why They Should Be Used and How They Are Being Used," *Theory into Practice*, vol. 47, no. 3, 2008, pp. 229–239.
14. L. Clemetson, "A Nation at War: The Carrier; A Diverse Crew Reflects the Nation's Social Changes," *The New York Times*, 30 March 2003.
15. M. Fox, "Appendix 1: Video Game Chronology," *The Video Games Guide*, 2nd ed. Jefferson N.C.: McFarland & Company, 2013, pp. 337–354.
16. M.J.P. Wolf, "A Brief Timeline of Video Game History," *The Video Game Explosion: A History from PONG to Playstation and Beyond*, Westport, Conn.: Greenwood Press, p. xvii.
17. A. Lenhart, J. Kahne, E. Middaugh, A. Macgill, C. Evans, and J. Vitak, "Teens, Video Games and Civics," Pew Research Center Report, 16 Sept. 2008, p. 2.
18. J. Simões, R.D. Redondo, and A.F. Vilas, "A Social Gamification Framework for a K–6 Learning Platform," *Computers in Human Behavior*, vol. 29, no. 2, 2013, pp. 345–353.
19. D. Oblinger, "The Next Generation of Educational Engagement," *Journal of Interactive Media in Education*, vol. 8, 2004, p. 14.
20. B. Marr, "Electronic Arts: Big Data in Video Gaming," *Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results*. Chichester, U.K: Wiley & Sons, 2016, pp. 273–279.
21. C.C. Abt, "The Reunion of Action and Thought," S*erious Games*. Lanham, Md.-London: University Press of America, 1987, pp. 3–14.
22. T. Mahnken, "Anti-Ship Cruise Missiles," *The Cruise Missile Challenge*, Washington, D.C.: Center for Strategic and Budgetary Assessments, 2005, pp. 9–13.
23. A. Feickert, "Missile Survey: Ballistic and Cruise Missiles of Foreign Countries," Congressional Research Service Report for Congress RL30427, 5 March 2004.
24. Naval Studies Board, "Cruise Missile Threats," *Naval Forces' Capability for Theater Missile Defense*, Report of the Naval Studies Board. Washington, D.C.: National Academy Press, 2001, pp. 26–30.
25. P. Zarchan, "Proportional Navigation and Weaving Targets," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 5, 1995, pp. 969–974.
26. N.F. Palumbo, "Homing Missile Guidance and Control," *Johns Hopkins APL Technical Digest*, vol. 29, no. 1, 2010, pp. 2–8.
27. D. Adamy, "Ship-Protection Model Example," *EW 101: A First Course in Electronic Warfare*. Norwood, Mass.: Artech House, 2001, pp. 248–250.
28. Naval Studies Board, "The Role of Hard-Kill and Soft-Kill Assets," *Integration of Hard-Kill and Soft-Kill Systems for More Effective Fleet Air Defense*. Washington D.C.: National Academies Press, 1992, pp. 16–19.
29. W.S. Carus, "Antiship Missile Defenses," *Cruise Missile Proliferation in the 1990s*. Westport, Conn.: Praeger, 1992, pp. 101–102.
30. M. Fuller, "Mk 15 Close-in Weapon System, Phalanx," *Jane's Naval Weapon Systems*, issue 55. Alexandria, Va.: IHS Global Limited, 2011, pp. 650–654.

31. D.G. Kiely, "Active Electronic Warfare—ECM," *Naval Electronic Warfare*, Brassey's Sea Power: Naval Vessels, Weapons Systems and Technology Series, Vol 5. London-Washington, D.C.: Brassey's Defence Publishers Ltd., 1988, pp. 50–77.
32. M. Streetly, "United States," *Jane's Radar and Electronic Warfare Systems*, 2011–2012," Alexandria, Va.: IHS Global Limited, 2011, pp. 150–164 & 465–469.
33. S. Thiagarajan, "The Myths and Realities of Simulations in Performance Technology," *Educational Technology*, vol. 38, no. 5, 1998, pp. 35–41.
34. T. Killion, "Future Naval Capabilities," presented at the *NDIA 15th Annual Science and Engineering Technology Conference*, College Park, Md., 9 Apr. 2014.
35. Chief of Naval Operations Directive, "Speed to Fleet Process," release from the Department of the Navy, OPNAVINST 3050.26, 9 Apr. 2015.
36. United States Navy, "U.S. Navy Program Guide 2015," Release from Department of the Navy, 2015.
37. C. Butler, "The Effect of Leaderboard Ranking on Players' Perception of Gaming Fun," *Online Communities and Social Computing*, A. Ant Ozok and P. Zaphiris, eds. Berlin-Heidelberg-London: Springer-Verlag, 2013, pp. 129–136.
38. J.J. Lee and J. Hammer, "Gamification in Education: What, How, Why Bother?" *Academic Exchange Quarterly*, vol. 15, no. 2, 2011, pp. 146–150.
39. M. Gombolay, "Machine Learning for Education: Learning to Teach," *IEEE Transactions on Computational Intelligence and Artificial Intelligence in Games*, Nov. 2016.
40. L. Curry, "An Organization of Learning Styles Theory and Constructs," presented at the Annual Meeting of the American Educational Research Association, Montreal, Quebec, Canada, 11–15 Apr. 1983, http://eric.ed.gov/?id=ED235185.
41. M. Sailer, J. Hense, H. Mandl, and M. Klevers, "Psychological Perspectives on Motivation through Gamification," *Interaction Design and Architectures Journal*, vol. 19, 2013, pp. 28–37.
42. J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the 5th Berkeley Symposium of Mathematics, Statistics, and Probability*, vol. 1, 1967, pp. 281–297.
43. P.J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Journal of Computational and Applied Mathematics*, vol. 20, no. 1, 1987, pp. 53–65.
44. T.K. Moon, "The Expectation-Maximization Algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, 1996, pp. 47–60.
45. L. Kaufman and P. Rousseeuw, "Clustering by Means of Medoids," *Statistical Data Analysis Based on the L1 Norm*, 1987, pp. 405–416.
46. D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 9, 1993, pp. 850–863.
47. H. Gardner, "Personalized Education: A Quantum Leap in Learning Will Allow Everyone to Go to the Head of the Class," *Foreign Policy*, vol. 172, 2009, p. 86.
48. A.W. Melton, *Categories of Human Learning*. New York: Academic Press, 1964.
49. L. E. Baum and T. Petrie, "Statistical Inference for Probabilistic Functions of Finite State Markov Chains," *The Annals of Mathematical Statistics*, vol. 37, no. 6, 1966, pp. 1554–1563.
50. M. Gombolay, R. Jensen, J. Stigile, S-H. Son, and J. Shah, "Apprenticeship Scheduling: Learning to Schedule from Human Experts," *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, 2016, pp. 826–833.
51. P. McDowell, "Strike Group Defender Assessment," to-be-published report for the Modeling, Virtual Environments and Simulation (MOVES) Institute at the U.S. Naval Postgraduate School, Monterey, Calif.

About the Authors

G. Mark Jones is a senior technical staff member in the Advanced Concepts and Technologies Group in the Air, Missile, and Maritime Defense Technology Division at Lincoln Laboratory. While at the Laboratory, he has worked as a modeler and systems analyst on projects including electronic warfare, threat capability assessment, and ballistic missile defense. Additionally, he has contributed to a range of innovative field tests. He holds a bachelor's degree in physics from the University of Kentucky and a doctoral degree in high-energy physics from the California Institute of Technology.

Matthew C. Gombolay held a postdoctoral position in the Ballistic Missile Defense System Integration Group at Lincoln Laboratory and has now assumed a position as the Catherine M. and James E. Allchin Early-Career Assistant Professor in the School of Interactive Computing at the Georgia Institute of Technology. He researches new computational methods to facilitate human-robot interaction and optimization, focusing on harnessing the strengths of human domain experts and sophisticated computational techniques to form collaborative human-machine teams for manufacturing, healthcare, and military operations. At the Laboratory and Johns Hopkins University Applied Physics Laboratory, he developed cutting-edge planning and scheduling algorithms for ballistic and anti-ship missile defense with the U.S. Navy and Missile Defense Agency. He received a Best Technical Paper Award from the American Institute of Aeronautics and Astronautics Intelligent Systems Committee, and his work has been highlighted in media outlets such as CNN, PBS, NBC, *Harvard Business Review*, and public radio. He holds a bachelor's degree from the Department of Mechanical Engineering at Johns

Hopkins University and master's and doctoral degrees from the Department of Aeronautics and Astronautics at MIT, where he was a National Science Foundation graduate research fellow in MIT's Interactive Robotics Group.

Reed E. Jensen is a technical staff member in the Advanced Concepts and Technologies Group. He has been with Lincoln Laboratory for more than 10 years and has worked in the areas of electronic warfare systems and missile system dynamics. His recent emphasis is on autonomous control and coordination of surface and air platforms. He holds a bachelor's degree in physics from Brigham Young University and a master's degree in control theory from Northeastern University.

Steven L. Nelson is a technical staff member in the Interceptor and Sensor Technology Group. After joining Lincoln Laboratory in 2002, he has been involved in work on the Aegis Ballistic Missile Defense (BMD) System. In 2004, he became the co-lead of efforts to create an integrated discrimination architecture, which resulted in the development of the Aegis BMD Discrimination Prototyping Environment, a version of which was successfully demonstrated on the FTM-12 flight test in 2007. Most recently, he has been involved in the Aegis Sea-Based Terminal efforts, concentrating on guidance and simulation. He holds a bachelor's degree in electrical engineering and computer science from MIT and a master's degree in electrical and computer engineering from Northeastern University.