Enemy Location Prediction in Naval Combat Using Deep Learning

Morgan Freiberg
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, U.S
msf6wk@virginia.edu

Kent McLaughlin
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, U.S.
kjm8zb@virginia.edu

Adinda Ningtyas
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, U.S
an5ka@virginia.edu

Oliver Taylor
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, U.S.
omt3bf@virginia.edu

Stephen Adams
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, U.S.
sca2c@virginia.edu

Peter A. Beling
Department of Engineering Systems and Environment
University of Virginia
Charlottesville, U.S.
pb3a@virginia.edu

Roy Hayes
Systems Engineering, Inc.
Dulles, U.S.
rhayesjr@seisecure.com

***Abstract*—The immensely complex realm of naval warfare presents challenges for which machine learning is uniquely suited. In this paper, we present a machine learning model to predict the location of unseen enemy ships in real time, based on the current known positions of other ships on the battlefield. More broadly, this research seeks to validate the ability of basic machine learning algorithms to make meaningful classifications and predictions of simulated adversarial naval behavior. Using gameplay data from World of Warships, we deployed an artificial neural network (ANN) model and a Random Forest model to serve as prediction engines that update as the battle progresses, overlaying probabilities over the battlefield map indicating the likelihood of the unseen ship being at each location. The models were trained and tested on gameplay data from a World of Warships tournament in which former naval officers served as commanders of competing fleets. This tournament structure ensured cohesive and coordinated naval fleet behavior, yielding data similar to that seen in real-world naval combat and increasing the applicability of our model. Both the Random Forest and ANN model were successful in their predictive capabilities, with the ANN proving to be the best method.**

***Keywords*—Artificial Intelligence, Intent Inference, Machine Learning, Naval Combat**

I. Introduction

Machine learning promises to glean insights from the mountains of data produced daily and to offer solutions to problems too complex for the human mind to solve alone or in real-time. Military affairs offer a host of exciting possible applications for machine learning and artificial intelligence (AI). The impact of the integration of machine learning into the military can save time, money, and most importantly, lives. Machine learning will provide valuable and previously unavailable information, allowing decision-makers to make quicker and better informed choices.

Naval combat is characterized by speed and ferocity, with precise coordination and cohesive strategy being of the utmost importance [1]. Quick decisions made in the turmoil of naval battle can have drastic repercussions on the billions of dollars of equipment and hundreds of lives involved in each engagement. As such, it is crucial for leaders to have access to the best actionable information possible at any point in time. AI, and machine learning in particular, offers a route to not only find valuable insights from naval battlefield data, but to do so in real-time, ensuring that in-the-field decision-makers are better equipped to make the most informed decisions. The integration of machine learning into the military is becoming more crucial by the day, as intensifying international competition has resulted in a veritable arms race for intelligent military systems [2]. AI and machine learning systems will yield a decisive advantage in this high-stakes domain.

The focus of this paper is to present foundational work for adversarial intent inference, namely predicting the location of unseen enemy ships based on the known location of other ships on the battlefield. This prediction capability will provide naval officers with a probabilistic means of assessing where an enemy ship is most and least likely to be, allowing them to make tactical decisions accordingly. More broadly, our research seeks to validate the ability of basic machine learning algorithms to draw meaningful and actionable conclusions based on information available in a combat scenario.

In lieu of using limited data from actual naval battles, we used data extracted from a naval warfare video game, *World of Warships*. *World of Warships* is a massively multiplayer online game wherein each player controls a single ship, with each battle consisting of two teams of equal size battling against one other. There are a variety of maps and gameplay objectives, but battles are largely focused on eliminating all enemy ships. While using this game as our data source limits our predictive model's immediate applicability, it allows us to use diverse data from a large number of simulated naval battles that are reasonably reflective of real-world engagements.

Following the data extraction process and parsing, we deployed both a Random Forest model and an Artificial Neural Network (ANN) for location prediction of an unseen ship. Both models effectively predicted the location of the target ship, but the ANN proved to be the superior model.

The next section provides a review of existing work in location prediction algorithms. We follow our review of related work with a detailed discussion of our approach and results. Lastly, we provide a brief conclusion and proposal for future work that will build on the results.

II. Related Work

While machine learning and AI are set to become increasingly prevalent in the military in the coming years, relatively little work has been done in applying machine learning techniques for predictive analysis in a dynamic combat scenario. The work in this realm that produces predictive models tends to focus on the location prediction of an object or event, rather than a mobile adversary's location.

Wei et al. used a game theoretic approach for predicting the likelihood of insurgent attacks at different locations [3]. Their spatial-temporal model made predictions based on area features and past attacks, accounting for the possibility that the insurgents would change their behavior pattern to achieve surprise attacks.

Lerner deployed an artificial neural network to predict the emplacement of improvised explosive devices (IEDs) using simulated past emplacements and a variety of features such as terrain and traffic volume [4]. While based entirely on simulated data, the model was able to achieve an impressive 86% predictive accuracy. Grant and Stewart similarly developed a model for predicting IED emplacement, but incorporated the additional complexity of IED device reliability as well as the human aspects of IED attack effectiveness [5].

Where location prediction research in the military domain has previously focused on predicting the location of events and objects, our research provides a continuously updating model for predicting the location of an active and mobile enemy vessel.

The proposed model provides crucial in-battle information to naval officers, presenting an evolving probability map identifying where an unseen enemy is most and least likely to be at any moment, given all available information. Such a model is in-line with other ongoing research into intelligent military systems, such as the augmented reality system proposed by multinational arms manufacturer, BAE Systems [6]. BAE's proposed system provides a heads-up display for a naval officer, overlaying valuable tactical information onto their real-world view. A real-time predictive map of enemy location, as proposed in our work, can integrate seamlessly into this augmented reality system as well as into the emerging intelligent military more broadly.

III. Approach

Data was obtained from five separate 6 v 6 *World of Warships* battles involving volunteers as ship operators and retired naval officers as fleet commanders. By using data from these officer-led teams rather than standard *World of Warships* multiplayer battles, we were able to more adequately replicate the coordination and strategy of real-world naval combat engagements. This more realistic battle format yielded data that more accurately represents real-world data, making our models more immediately useful and applicable.

Gameplay data from *World of Warships* is automatically saved to a replay file. The gameplay data is encrypted under the Blowfish Algorithm and as a result needed to be parsed. A publicly available replay parser under an MIT license was used as a base for analysis [7] . This parser was modified for the metrics appropriate to our research. These alterations are publicly available in an MIT licensed GitHub repository forked from Shyshatsky's replay parser [8]. Our previous paper provides additional details about the data parser and data processing methodology [9].

The process of extracting gameplay data from a video game for the purpose of machine learning model-building is similar to that used by Hayes and Beling in their study involving *StarCraft: Brood War* [10]. Their work served as a reference point for our initial research process, but where they used unsupervised hierarchical clustering for opponent behavior modeling, we used an ANN specifically for opponent location prediction.

A. Replay Combination

Each replay file contains an incomplete view of the game, holding only the data necessary to replicate the game from the point of view of the owner of the replay. As a result, four replays from each of the games in the tournament, two from each team, were saved and processed in order to recreate a complete view of each game. The replay files were parsed to extract timestamped location data, as well as a few ship and map characteristics such as ship type, team identification, position of gameplay objectives, and map size. For each game, the replay location data was combined and location was taken for every second of a game, yielding a complete picture of the game state for every timestamp.

B. Data Restructuring

The goal of the model is to predict the presence of an unseen target ship. Given the limited amount of battle data available, the model could not be dependent on the fixed position of ships. In order to define standard prediction parameters, each map was split into a 10x10 grid based on the physical dimensions of the map. The cardinal coordinate positions from the replay data for each ship were then mapped to the corresponding cell in this 10x10 grid.

Each entry in the grid contains an array of model features as well as a corresponding binary dependent variable value indicating if the target ship is within that cell. The features used in the model, as detailed in Table. 1, include a mix of target ship characteristics and distance measures as related to the remapped gridded positions. One feature used in the model is the type of the target ship using a one hot encoded representation of the three ship types (Destroyer, Cruiser, and Battleship) with Destroyer as the base case. The other features are all different distance measures including the distance of the cell to the center of the map, the distance to the nearest enemy ship, the distance to the nearest friendly ship, and the distance to the nearest scoring objective. The target ship was excluded from consideration within these distance measures as to not include any information about the location of the ship.

TABLE I. Model Features

Variable Name	Description
isCruiser	binary value indicating if target ship is a Cruiser (1) or not (0)
isBattleship	binary value indicating if target ship is a Battleship (1) or not (0)
centerDist	distance from the current grid square to the center of the grid
controlPointDist	distance from the current grid square to the closest control/objective point
enemyDist	distance from the current grid square to the closest enemy ship
friendlyDist	distance from the current grid square to the closest friendly ship (excluding the target ship)

In order to expand the data and make a model that is more extensible to other games and ships, every ship in a given battle was rotated through the data structuring process as the target ship, such that each battle provided a distinct search for each of the 12 participating ships. Each trial ended with either the destruction of the target ship or the end of the battle, whichever happened first. The restructuring of the data into the grid structure created severely imbalanced target classes, with a single target value of 1, indicating the location of the target ship, for every 99 value instances of 0, indicating the target ship is not in that cell. To help alleviate this issue, the training data was selectively sampled to help balance the classes. First, the data was undersampled by taking random subsamples of the instances corresponding to the target value of 0 to reduce to 75% the original size. The data was subsequently oversampled by copying all instances of the 1 class twenty times. This resulted in 20 instances of the 1 class for about every 75 instances of the 0 class. This processing was done for all five games with 3 games held for the training set, 1 for validation, and 1 for testing, with the exception of sampling which was only performed on the training games. A breakdown of the amount of data is shown in Table. 2.

TABLE 2. Gameplay Data Usage

	Training	Validation	Testing
Number of Games	3	1	1
Number of Grids	18,234	5,154	6,537
Sampled?	Yes	No	No
Total Number of Data Entries Used	1,736,788	515,400	653,700

C. Model Architecture and Training

A feed-forward network was constructed in PyTorch using linear layers with ReLU performed between each layer. Drop out layers were used to combat overfitting and the final output was calculated using the sigmoid function. The full architecture of the model is illustrated in Figure 1. Since little was known about the possible relationships between the features and the location of the target ship, the model began with 4 large exploratory layers. These layers were followed by 11 consecutive, equally sized layers with the intuition that each ship, or features related to individual ships, could be encoded in their own layers.

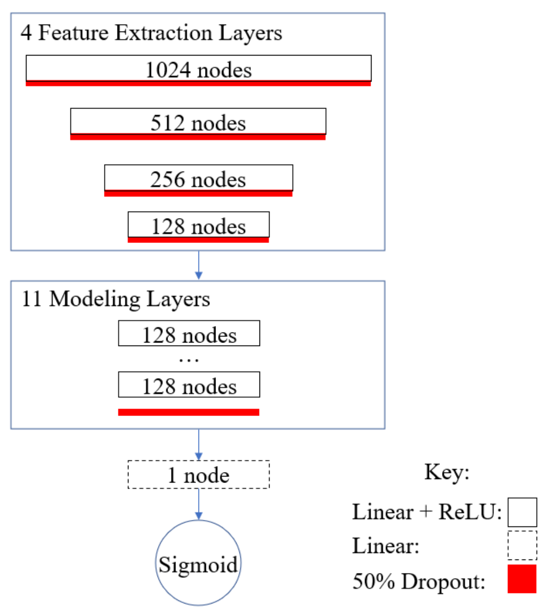

Fig. 1. ANN architecture.

Training was done through batch training with a batch size of 128 using Binary Cross Entropy loss and an Adam optimizer. The model was trained with 100 epochs and the area under the ROC curve was used as the metric of performance. The performance of the model throughout training is shown in Figures 2 and 3 which outline the BCE Loss and AUC for the training and validation data for each epoch. The epoch that produced the highest validation AUC, in this case epoch 15, was chosen for the final model.

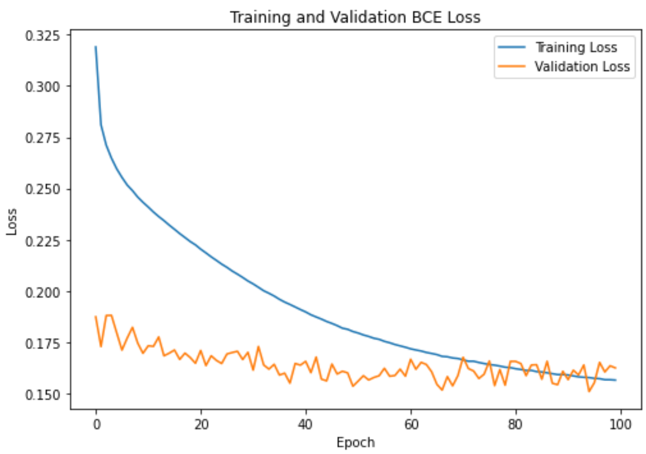

Fig. 2. Training and validation BCE Loss throughout training.

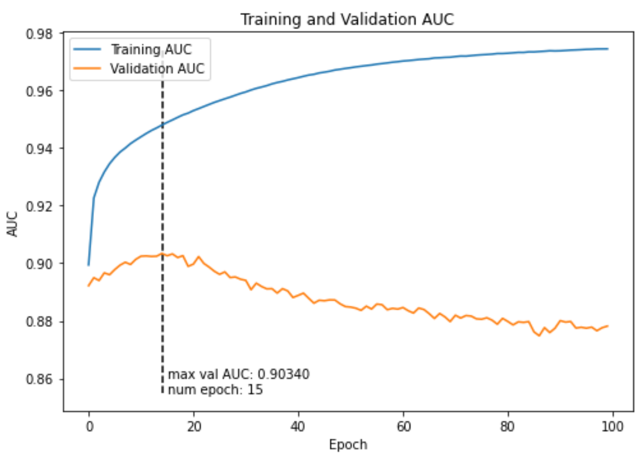

Fig. 3. Training and validation AUC throughout training.

IV. Results

The final ANN and Random Forest models are listed in Table 3. Both models performed well under the AUC performance metric. Additionally, the ROC curves for the ANN model are shown in Fig. 4, illustrating generally excellent results for the model.

TABLE 3. Model AUCs

	Data Set		
	Training	Validation	Testing
ANN	0.949	0.903	0.824
Random Forest	0.932	0.911	0.764

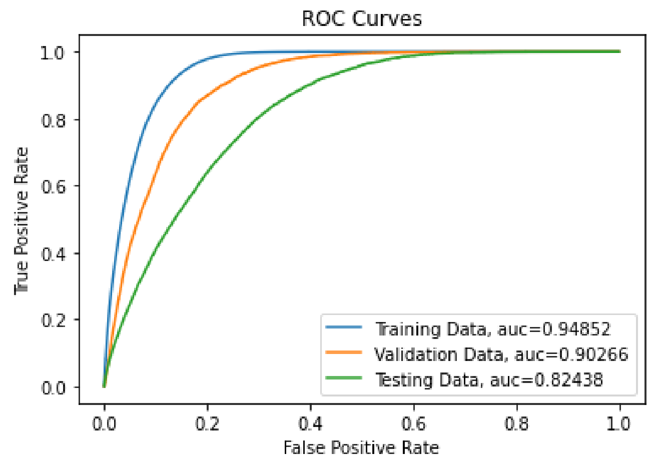

Fig. 4. ROC Curves for the ANN model trained for 10 epochs

For the point of comparison, a Random Forest classification model trained and tuned on this data, resulting in a model with 10 estimators and a maximum depth of 5. The ROC curves for the Random Forest, which were found to be comparable to the deep learning model, are seen below in Fig. 5

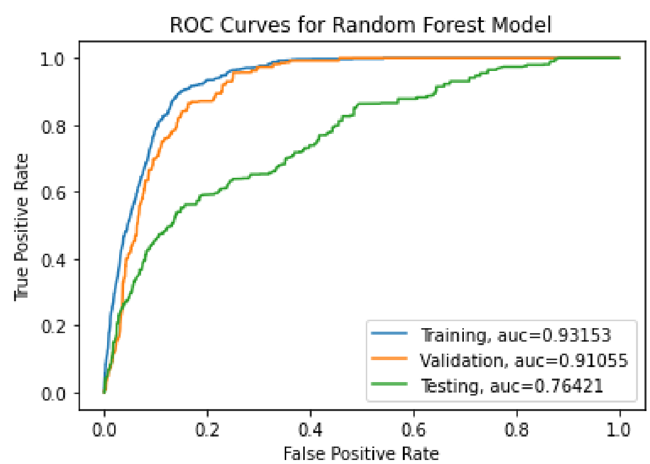

Fig. 5. ROC Curves for the Random Forest model.

The comparable performance of the Random Forest model to the deep learning model suggests that there exists clear relationships between the features of the model and the presence of the target ship in a cell. The noticeable drop of 0.06 in the testing set suggests that the Random Forest model is less extensible to data outside of the training and validation sets, but more game data is needed to fully quantify the degree to which this is true. However, the high AUC values of the Random Forest model suggests that the more complex deep learning approach is not required. It is important to consider that the model is based on data from a video game that simplifies real world situations, so deep learning should not be immediately ruled out. Furthermore, the inclusion of more gameplay data or identification and calculation of additional features may require a deep learning approach.

The outputs of the ANN model may be overlaid on the game map to see the predicted probabilities of the target ship being within each cell. An example output of the model performing correctly overlaid on a fully detailed gameplay map is shown below in Fig. 6. The map includes the location of friendly and enemy ships, as well as the objective points A, B, and C. The level of opaqueness indicates the predicted probability of the target ship occupying that square in the grid.

Fig. 6. Model location predictions overlaid on a detailed gameplay map. Green ships are friendly, red are enemy; the yellow outlined star is the unseen enemy target ship.

To create this visual, the predictions of the individual square probabilities were regrouped into their respective grids. In doing so, the model can be assessed by the accuracy of the N highest probability squares on the grid at each timestep. Table 4 indicates the percentage of correct predictions of the target ship location when including the top 1, 5, and 10 probability squares on the grid.

TABLE 4. Top N Accuracies (%)

		Data Set		
		Training	Validation	Testing
N	1	18.27355	3.58945	6.68502
	5	59.21904	34.26465	21.76840
	10	82.74103	62.98021	37.72373

These accuracies illustrate a clear advantage of the model over guessing, indicating the model can, to some degree, successfully predict the location of the target ship. The drop in accuracy from the training to validation and testing sets may indicate overfitting to the training data, however this may also be due to the limited number of games available for training. The large drop between validation and testing may also point to the limited data used as the gameplay may vary widely between battles and since no data from the testing game is included in training, the model may be unaware of new patterns that emerge. In either case, the model successfully identified meaningful relationships between the features and the location of the target ship that were extensible to game play outside of seen data. The successful construction of this model provides a solid foundation for and a convincing argument to pursue future research.

Discussion

The ANN model yielded very promising results in its ability to identify one-tenth of the map as the location of the unseen ship with almost 40% testing accuracy. While by no means a perfect prediction engine, this level of predictive accuracy achieved with only 3 training battles already serves to provide a fleet commander with valuable information about the highest likelihood position of an unseen enemy vessel. Perhaps equally important is the ability of this model to reliably predict where an unseen enemy ship is *not* located, which is tremendously helpful information for avoiding ambushes or other forms of surprise naval attack.

While falling short of some of the predictive models for IED emplacement and attack locations discussed in Related Work in terms of pure predictive accuracy, this model updates in real-time in response to multiple mobile enemies on a constantly-evolving battlefield. The performance of the model could be improved by increasing the amount of available data through more tournament play, perhaps with an emphasis on including multiple strategies throughout play. Additional features, such as encoding geographical landmarks on the maps, would also prove valuable to the model. With the availability of more training data and a richer set of features from an enhanced military simulator, our promising ANN model can be refined further to provide a crucial advantage to a decision-maker in the chaotic heat of battle.

V. Conclusion

The increasing prevalence of machine learning and other AI technology in military affairs will allow decision-makers in combat situations to make informed, data-driven decisions in a way that has never before been possible. The model created in this research illustrates that location prediction of unseen ships in the dynamic domain of naval warfare setting is not only possible, but can yield a decisive advantage. Quality information is of vital importance in naval warfare, and the predictive model developed here can serve as an essential guide to a naval commander working from limited available information.

While our model is constructed for use with *World of Warships* gameplay data, the framework, which is based solely on information that is consistently available in naval combat, can be seamlessly adopted for real-world application. The commander standing on the bridge of their ship no longer has to rely on an educated guess in assessing where a deadly ambush may be lurking. This research has proven that our ANN model and other machine learning techniques can provide a real-time map of information on an adversary that has yet to be spotted.

A second major contribution of this study is the foundational ability of machine learning algorithms to successfully leverage available data to drive live-combat decisions. While there is much trepidation surrounding the integration of intelligent systems into the military for fear of losing the moral human element of decision-making, our research provides an example of humans being kept in-the-loop on an intelligent military system. Not only will AI and machine learning provide a strategic advantage on the battlefield, but this advantage can be obtained without sacrificing command control to computer systems. Our aim for this study is to provide a framework for delving into naturally emerging combat behavior patterns to reveal valuable and actionable insights that are not readily apparent to human decision-makers in the heat of battle.

Acknowledgements

We would like to thank Systems Engineering, Inc. for their sponsorship of our research and guidance throughout. We are also grateful to Rear Admiral Guy Curtis (USN retired) and Captain George Jackson (USN retired) for their expertise and guidance in commanding the teams for the *World of Warships* tournament. Finally, we would like to thank the student volunteers for their time and energy given to the tournament, which provided essential data for our research.

References

[1] D. Knox, "The role of doctrine in naval warfare," in United States Naval Institute Proceedings, vol. 41, pp. 325–366, 1915.

[2] W. Wang, H. Liu, W. Lin, Y. Chen, and J. Yang. "Investigation on works and military applications of artificial intelligence," in IEEE Access, vol. 8, January 2020, pp. 131614–131625.

[3] M. Wei, G. Chen, J. Cruz Jr., L. Haynes, and M. Kruger, "Applying spatial-temporal model and game theory to asymmetric threat prediction," United States: Office of Naval Research, Conference Paper, 2007.

[4] W. Lerner, "Predicting the emplacement of improvised explosive devices: an innovative solution," Capitol College: Ph.D. Dissert., J. Pittman, Adv., 2013.

[5] M. Grant and M. Stewart, "A systems model for probabilistic risk assessment of improvised explosive device attack" in Int. J. of Intelligent Defence Support Systems, vol. 5, no. 1, pp. 75–93, 2012.

[6] P. Donaldson, "Deep learning: Artificial intelligence for naval forces," Naval Forces, vol. 40, no. 1, pp. 13–15, 2021.

[7] O. Shyshatskyi, "Replay Parser (WoT & WoWS)," GitHub Repository, https://github.com/Monstrofil/replays_unpack (accessed Sep 6, 2020).

[8] M. Freiberg and O. Shyshatsky, "World of Warships Replay Parser (forked from Monstrofil/replays_unpack)," GitHub Repository, https://github.com/m-freiberg/replays_unpack (accessed Sep 6, 2020).

[9] A. Ningtyas, P. Beling, O. Taylor, M. Freiberg, K. McLaughlin, et al., "Machine Learning Classifications of Player Behavior," unpublished, SSRN, 2021.

[10] R. Hayes and P. Beling, "Unsupervised hierarchical clustering of build orders in a real-time strategy game," The Computer Games Journal, vol. 7, no. 1, pp. 5–26, 2018.