Defence Science Journal, Vol. 75, No. 2, March 2025, pp. 179-187, DOI : 10.14429/dsj.20246

MOP-N: A Hybrid AI Model for Automated Deployment of Guns in Dynamic Wargaming Scenario

S.B. Taneja[#,*], Sourabh Jaiswal[#], Sanjay Bisht[#], Vinita Jindal[^] and Punam Bedi[$]

[#]*DRDO-Institute for Systems Studies & Analyses, Delhi - 110 054, India*
[^]*Keshav Mahavidyalaya, University of Delhi, Delhi - 110 034, India*
[$]*Department of Computer Science, University of Delhi, Delhi - 110 007, India*
[*]*E-mail: sbtaneja@yahoo.co.in*

ABSTRACT

Wargames play a critical role in training of military officers/commanders for operations planning and execution. A typical training wargame may involve a large number of players as part of more than one competing forces. It is often desirable to reduce number of players to improve playability of wargames, which is in conflict with the requirements of operational realism. Successful automation of players and their decision making processes can result in reduction of number of players without loss in the operational realism. Also, depending upon the objective, a wargame can be played at various levels of abstraction viz. procedural, tactical, operational and strategic. Usually higher level (operational and strategic) wargames involve abstract (low resolution) entities. Developing models to simulate these abstract entities is most critical challenge for wargaming simulation designers. Alternatively, with incorporation of automation of lower level players' decision making processes, detailed high resolution entities can be represented enabling modelling at abstraction closer to real physical entities, which is relatively simpler. The automation may be required at different stages of a wargame. It may be required during game initialisation or in relatively static situations where time constraint on decision making (and thus computation time) may not be severe. In contrast, decision making in dynamic situations may put severe time constraint on the automated decision making and computation time. Current work explores mechanism to automate the players' decision making processes in time constrained scenarios. Specifically, problem of deployment of guns in the mountainous terrain has been considered. MOP-N a hybrid approach comprising of Monte-Carlo Simulation, Predator-Prey Particle Swarm Optimisation (PP-PSO) algorithm and Neural Network (NN) has been used for automated deployment of guns w.r.t. adversary deployments, terrain and road network in time constrained situations. Here, Monte-Carlo simulation is used to generate feasible adversary deployment scenarios, PP-PSO algorithm is used to generate corresponding own gun deployments (in time relaxed manner) and this data is used to train the NN in prediction of own gun deployments in time constrained situations. Results show that the machine learning model has high effectiveness as generated results are within 8 % of the PP-PSO results, whereas the computation time decreased by more than 99.6 %. The approach has immense potential in various automated deployment problems of military simulation and decision support domain.

Keywords: Machine learning; Artificial intelligence; Particle swarm optimisation; Gun deployment; Wargames; Constructive simulation; Military decision support

1. INTRODUCTION

Computerised wargaming is a practice used by the military to train commanders to plan and analyse various operations against given dynamic threat in its surrounding environment. The objective is to improve and assess commander's planning capabilities in fog of war situations. Wargames involve compressed time simulations wherein commanders take decisions in a military scenario they seek to shape; rules determine scope of the decisions made by them and models determine results that specify how decisions and subsequent results affect environment and military elements[1].

In computer wargames, there is a need to automate many of the commander's decision-making processes, so that the effectiveness of these wargames in training commanders is enhanced. First, to reduce number of participants in a constructive compressed time simulation-based wargame, there is a need to have automated elements making instantaneous decisions taking various operational/tactical factors, environment and threat into consideration. Second, for high-fidelity simulation with abstract input, there is a need to automate the orders given by higher level commanders into smaller multiple orders for sub-ordinate units' commander. Though simulation accepts a higher abstraction input, simulation progresses at higher resolution generating higher-fidelity results instead of lower fidelity results which would have been generated with use of abstract simulation.

The automated decision making may be required during game initialisation or in relatively static situations (scenarios elements are relatively static in the given temporal and

Received : 27 May 2024, Revised : 07 October 2024
Accepted : 28 October 2024, Online published : 24 March 2025

spatial space of the battle situation), where time constraint on decision making (and thus computation time) may not be severe. In contrast, during the intense dynamic phase of the battle simulation where the enemy may change its posture and locations, frequent and fast deployment of own guns is required as counter-action. In these battle situations, the computation time for the automated deployment becomes important for the solution being proposed.

We, in our previous works[2-4] have suggested PSO/APSO/PP-PSO algorithm based approaches for solving such problems during game initialisation/ static situations. In current paper, a conceptual framework, MOP-N (comprising of Monte-Carlo Simulation, PP-PSO algorithm and Neural Network) has been proposed for automation for deployment class of decision-making problems in dynamic situations. Specifically, problem of deployment of own indirect fire guns in a mountain warfare scenario has been considered. Due to limited availability of real-world data, Monte-Carlo simulation-based approach has been used for generation of training, validation and test datasets. Neural Network trained on the generated data has been used for prediction of deployment locations of own guns.

The rest of the paper is organised as follows. Section 2 provides a summary of the related work regarding deployment site prediction algorithms and automated decision making in wargames. It is followed by detailed description of the problem, solution approach, experimental setup, results and analysis. Lastly, conclusion and scope/suggestion for future work has been provided.

2. LITERATURE REVIEW

Though military application of machine learning, and AI has witnessed significant advances, applications related to use of machine learning techniques for making constructive simulation-based tools time efficient in a dynamic combat scenario are limited[5]. In case of military deployment class of problems, general research focus has been in application of heuristic algorithms for solution of these problems. Bisht[2], *et al.* established use of Particle Swarm Optimisation (PSO) and Adaptive Particle Swarm Optimisation (APSO) algorithms for military unit deployment in mountains considering terrain, adversary sites and various operational/tactical considerations. Infantry major weapon deployment problem in context of mountainous terrain has been solved using Predator-Prey PSO[3]. Comparison of performance of PSO, APSO and PP-PSO has been performed w.r.t. artillery deployment problem in mountain terrain[4]. The three problems consider a static scenario i.e., adversary locations are relatively static w.r.t. time. Gao[6], *et al.* use the Hybrid Particle Swarm Algorithm and hill-climbing method for solving the Location Problem of the Distribution Centre. The problem of deployment of multiple types of anti-air weapons has been addressed by Liu[7], *et al.*, where Memetic algorithm that uses a hybrid approach combining genetic algorithm and hill climbing has been used. Particle Swarm Optimisation (PSO) algorithm has been used by Jia[8], *et al.* to determine the optimal deployment scheme of defence force consisting of military units. Chen[9], *et al.* solved problem of deployment of firing units of networked air defence system. It uses artificial potential field method to optimise the deployment sites for the units minimising adversary vector penetration probability. Deep Kuhn–Munkres algorithm has been proposed by Sun[10], *et al.* for determining optimal deployment sites of single type of multiple weapon system in protection of vital assets against an incoming ballistic missile. The algorithm maximises the total available interception time for the incoming missiles. Missile fire allocation problem to given targets has been solved using Genetic Algorithm by Feng[11], *et al.*; the algorithm determines the attack scheme that maximise the damage efficiency. Peng-jiao[12], *et al.* proposed use of memetic algorithm to construct feasible deployment schemes for air defence weapon systems minimising penetration probability of enemy vectors. It uses queuing theory to estimate the penetration probability. Deep learning has been used by Freiberg[13], *et al.* for prediction of enemy ships' locations, based on positions of other known ships in vicinity. Gameplay data from a naval warfare game has been used to train the network. The network generates overlays in terms of probabilities of unseen ships' presence over each location in the given map.

3. PROBLEM DESCRIPTION

Deployment of forces is an important aspect of military decision making and in mountainous terrain it becomes even more important. Multiple factors like dynamic situation, phase of war, adversary posture, terrain, weather, deployability, approachability, survivability, effectiveness need to be taken into consideration while determining the optimum deployment site. As described in related work section, multiple attempts have been made to solve various types of military deployment problems using heuristic optimisation algorithms. These methods often require many iterations to arrive at a solution resulting in finite amount of computation time and do not generate results on near real time basis. However, due to battlefield dynamics, location of adversary units can change rapidly with time and thus there is a constraint on computational time requirement. The automated decision making need to be quick during abstract battlefield simulations and heuristic optimisation-based methods cannot be considered time efficient as such. In this context, current research aims to establish automation approach/method which can give good enough solutions in near real time.

For current purpose, problem of deployment of own indirect fire guns in mountainous terrain has been taken up. There can be multiple adversary units (say n) deployed. There is an operational requirement that m own guns per targets need to be deployed. Thus, total of $m \times n$ guns need to deployed. Commanders have to take into account multiple factors while deploying the guns in context of the given situation and operational/tactical objectives. The gun deployments are affected by following factors:

- Dominance of heights, terrain slope & gradients
- Visibility to own and adversary units
- Operational factors like logistics supply lines
- Guns capabilities factors such as their ranges, trajectories etc.

Weightage and relative importance of these factors can change with the changing situations. In our previous work[4], we have proposed a Predator-Prey Particle Swarm Optimisation (PP-PSO) based algorithm to solve the problem catering for adversary unit's deployment sites, road network and Digital Elevation Model (DEM) of surrounding terrain. The PP-PSO algorithm maximises the effectiveness and survivability of own guns in terms of a Measure of Effectiveness (MoE) defined as weighted sum of various operational factors. The solution considered a static scenario, where locations of adversary units remain unchanged with time; and thus, once calculated the results are always valid. The algorithm takes few hundred iterations to generate results requiring computation time of few minutes for a practical sized problem.

Current research extends our previous work[4]. The objective of the research is to establish an approach for use of machine learning models for solving the above problem in near real time. MoE as defined in our previous work[4] along with computation time requirement is used to characterise the solutions predicted by neural network for ascertaining the efficacy of proposed methodology.

4. METHODOLOGY

The current problem is modelled as a multi-response regression problem. The locations of adversary deployment sites and related parameters become the independent variables for the regression and the locations of own guns become the response. A non-linear regression model is presumed in between. Since there are multiple guns, whose locations need to be determined, it becomes a multiple-response problem. Since neural networks are well known to act as complex non-linear regression model when working with numeric input and response data, current research utilises the neural network to learn the input-output relationship. Location of adversary deployment sites, road network, and DEM data is fed to the proposed model to predict the location of own gun sites as the response. Training of the proposed model requires a large amount of training data representing different patterns that may exist between input and the response variables. As availability of real-world data is limited, it is proposed to use Monte-Carlo simulation to generate large number of feasible adversary deployment sites. Output from Monte-Carlo simulation is used to generate input data for PP-PSO algorithm based automated gun deployment methodology[4]. Combination of Monte-Carlo simulation and PP-PSO generate the data of possible adversary locations and own gun deployments for the NN training. Details such as target probable area, no. of adversary deployment sites, own gun's probable area for deployment, DEM data, road network, and gun range & trajectory are utilised during the training and test data generation.

Figure 1 depicts the proposed approach in two parts. Figure 1(a) represents the activity sequence and inputs during the data generation and training phase of proposed approach, whereas Fig. 1(b) represents the activity and inputs during the exploitation (after NN deployment) phase. Subsequent sub-sections detail on the training data generation process and neural network architecture.

4.1 Training Data Generation

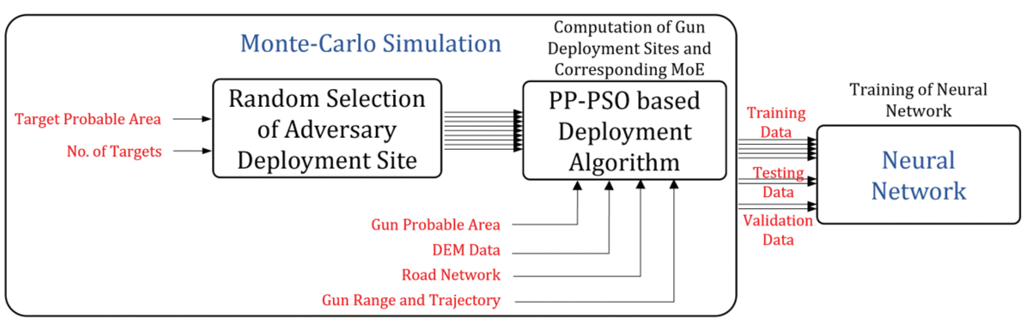
Simulation supported by automated deployment methodology[4] is used to create the training dataset. The terrain is loaded along with general deployment areas (in terms of polygons) for adversary & own forces and road network (as collection of points). Monte-Carlo simulations are used to generate adversary deployment sites. For each of the simulation run, required number of adversary deployment sites are randomly generated (uniformly distributed) within the adversary general area. PP-PSO based model[4] is used to generate corresponding own gun deployment sites for each set of adversary deployment sites.

PP-PSO is an algorithm based on the PSO algorithm and incorporates concepts of predator pouncing on the prey. In PP-PSO, predator follows best performing particle of the swarm. The update equations for velocity and position updates of predator are[3]:

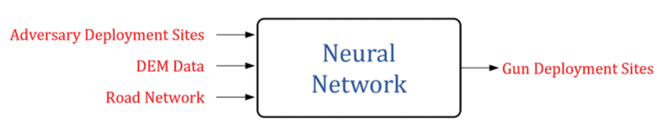
$$V_p(t) = c_4\left(X_g(t-1) - X_p(t-1)\right) \tag{1}$$

$$X_p(t) = X_p(t-1) + V_p(t) \tag{2}$$

Here, $V_p(t)$ is predator velocity at time instant t, $X_p(t)$ is



(a)



(b)

Figure 1. MOP-N approach for automated determination of gun deployment sites (a) Approach for pre-processing during NN training; and (b) NN exploitation after deployment.

predator position at time instant t, c_4 is predator velocity scale factor (a random number within the range 0 & some positive number), X_g and X_p represent best and normal positions of the predator. Swiftness/speed of the predator in catching a prey is dependent on selection of upper limit of c_4. Predator pouncing results in change of prey speed in one of the available dimensions. In absence of predator pounce, following equations establish velocity and position updates for prey:

$$v(t) = wv(t-1) + c_1 r_1 \left(p_l - x(t-1)\right) + c_2 r_2 \left(p_g - x(t-1)\right) \tag{3}$$

$$x(t) = x(t-1) + v(t) \tag{4}$$

where, $v(t)$ is particle velocity at time instant t, $x(t)$ is particle position at time instant t, w is inertia weight to control momentum of the particle, c_1 & c_2 (positive numbers) are acceleration constants. r_1 & r_2 are uniform random numbers in interval [0,1], p_l is the best solution for each individual

Table 1. MoE operational factors[4]

Operational factor and parameter	Mathematical formulation	Description
Road connectivity factor (O_1)	$O_1 = \frac{1}{n}\sum_{i=1}^{n} c_i$ N is number of FU c_i is link of ith FU from base supply road such that $c_i = \begin{cases} 1, & distance\ of\ i^{th}\ gun\ from\ supply\ road < dist_{min} \\ 0, & otherwise \end{cases}$ $dist_{min}$ is given minimum distance	The fire unit must be reachable/ movable by road/ track
Weapon range coverage factor (O_2)	Weapon range coverage function $O_2 = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}$ r_{ij} is distance of i^{th} FU from j^{th} target $r_{ij} = \begin{cases} 1, & r_{ij} < R_i \\ 0, & otherwise \end{cases}$ R_i is range of the artillery gun m is total number of targets to be fired	All enemy positions must be within the range (r) of the artillery gun
Crest clearance factor (O_3)	$O_3 = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} \delta_{ij}$ $\delta_{ij} = \begin{cases} 1, & \text{if crest clearance} \\ 0, & otherwise \end{cases}$	Artillery gun trajectory should not be obstructed by terrain profile
Reverse slope factor (O_4)	$O_4 = \frac{1}{nmH}\sum_{j=1}^{n}\sum_{k=1}^{m} s(z_j, z'_k)$ $z_j \equiv (x_j, y_j)$ is location of j^{th} FU $z'_k \equiv (x_k, y_k)$ Is location of k^{th} target $s(z_j, z'_k)$ Is difference of height of the j^{th} FU and height of point on line joining z_j and z'_k at some distance $d > 0$ from z_j; H is max height of designated deployment zone	Guns must get deployed over the reverse slope with respect to the enemy positions/ location/ axis
Terrain slope factor (O_5)	$O_5 = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} g_{ij}$ g_{ij} is average gradient of i^{th} FU w.r.t. j^{th} target such that $g_{ij} = \begin{cases} 1, & 0 \le \text{avg grad} < 15° \\ 0.4, & 15 \le \text{avg grad} < 30° \\ 0.2, & 30 \le \text{avg grad} < 45° \\ 0, & \text{avg grad} > 45° \end{cases}$	Gun must be deployed on the min admissible terrain slope/gradient

Table 2. Training data

S. No.	Adversary location 1		Adversary location 2		…	Adversary location 5		Own gun 1 location		Own gun 2 location		…	Own gun 15 location		No. of iterations	MoE
	X	Y	X	Y	…	X	Y	X	Y	X	Y	…	X	Y		
1	12927	16504	8538	10292	…	19577	6830	18383	5447	13559	5678	…	19782	5580	204	0.90922
2	9475	18065	6600	16011	…	19639	6889	13192	1799	17419	2036	…	14719	2784	186	0.913847
3	8913	18065	7752	13544	…	20813	3872	17242	8258	14098	6121	…	12511	3693	176	0.894184
4	10709	17186	7832	13809	…	20949	3561	19735	5576	18385	5453	…	19621	6545	189	0.912223
.	.	.	.	.		.	.	.	.	.	.		.	.	.	.
.	.	.	.	.	…	.	.	.	.	.	.	…	.	.	.	.
.	.	.	.	.		.	.	.	.	.	.		.	.	.	.
1398	12475	11404	15415	15527	…	20972	3558	14730	2768	17838	5259	…	21066	3470	176	0.915504
1399	20025	15117	10970	17773	…	8858	2739	14100	6074	15313	4321	…	16692	1921	187	0.896859
1400	12413	15872	9356	16757	…	10478	2084	17865	5239	17929	3671	…	17934	3669	200	0.91457

particle (local best) and p_g is the best solution from the whole population (global best).

In presence of predator-pounce, following equations govern prey velocity and position updates:

$$\begin{aligned} v(t) = {} & wv(t-1) + c_1 r_1 \left(p - x(t-1)\right) \\ & + c_2 r_2 \left(p_g - x(t-1)\right) \\ & - r_3 a e^{-bd} \end{aligned} \tag{5}$$

$$x(t) = x(t-1) + v(t) \tag{6}$$

Last term in velocity update equation above represents predator's repulsive effect on the prey. Here, a represents maximum amplitude and Euclidean distance between predator and prey (d) is controlled through parameter b. r_3 is uniform random number in interval [0,1].

Gun deployment problem is governed by multiple factors. A composite Measure of Effectiveness (MoE) which is weighted sum of the five factors (Table 1) is used to generated the solution. The composite MoE represents the overall efficacy of the generated solution. The unit commander decides about the relative weightage of individual factors based on his appreciation of the scenario and role of his unit. The details of the factors can be found in our previous work[4]. The PP-PSO objective function is:

$$\begin{aligned} & Max\, O = \sum_{j=1}^{5} w_j O_j \\ & 0 \le w_j \le 1, 0 \le O_j \le 1 \end{aligned} \tag{7}$$

where, O is weighted sum of the factors, O_1 to O_5 are the MoE factors and W_1-W_5 are the relative weightages given to the factors.

The results corresponding to large number of Monte-Carlo runs are generated and recorded. For each Monte-Carlo Run following data is recorded:

- Location of adversary deployment sites (in terms of cartesian co-ordinates of the sites)
- Location of own gun deployment sites (in terms of cartesian co-ordinates of the sites)
- Number of PP-PSO iterations for result generation
- Measure of Effectiveness (MoE) for the solution of simulation run

Table 2 provides a sample of the data record (for case when there are 5 adversary sites, 15 own guns and 1400 Monte-Carlo runs are performed):

4.2 Neural Network Architecture

Architecture of the proposed neural network model is depicted in Fig. 2. Neural network consisting of fully connected layers with three branches of input is used in current work. Depending upon the type of input with which the branch is working with, individual branches have additional layers, either fully connected or convolution. Inputs to the three branches of the network are:

- 5 adversary deployment sites (10×1 feature vector, 5 values for X co-ordinates and 5 values for Y co-ordinates of the deployment sites). The input is followed by a 10 node fully connected layer.
- Road network as set of 24 points (48×1 feature vector, 24 values for X co-ordinates and 24 values for Y co-ordinates of the road points). The input if followed by a 48 node fully connected layer.
- DEM data (648×792 matrix): Important terrain features are identified using convolution layer (filter size 5×5, stride 1 and 16 filters), batch normalisation and ReLU layers. Subsequently, using a fully connected layer, terrain is transformed into 50×1 feature vector. This 50×1 feature vector represents the terrain information for incorporation in the regression model.

All the three inputs are concatenated to form a 108×1 feature vector, which passes through multiple fully connected layers to generate the output feature vector of size 30×1, which is then rearranged into 15×2 matrix, representing 15 own gun deployment sites.

5. EXPERIMENTAL SETUP

MATLAB platform is used to implement the algorithm. Terrain DEM data (.geotiff) is loaded and displayed to the user (unit/sub-unit commander). A scenario in a mountainous terrain of size 19440m×23760m (represented by 648×792 DEM of 30-meter resolution) with vector features of road/tracks (as set of points in cartesian co-ordinates) has been taken up. User marks the general deployment areas (in terms of polygons) for

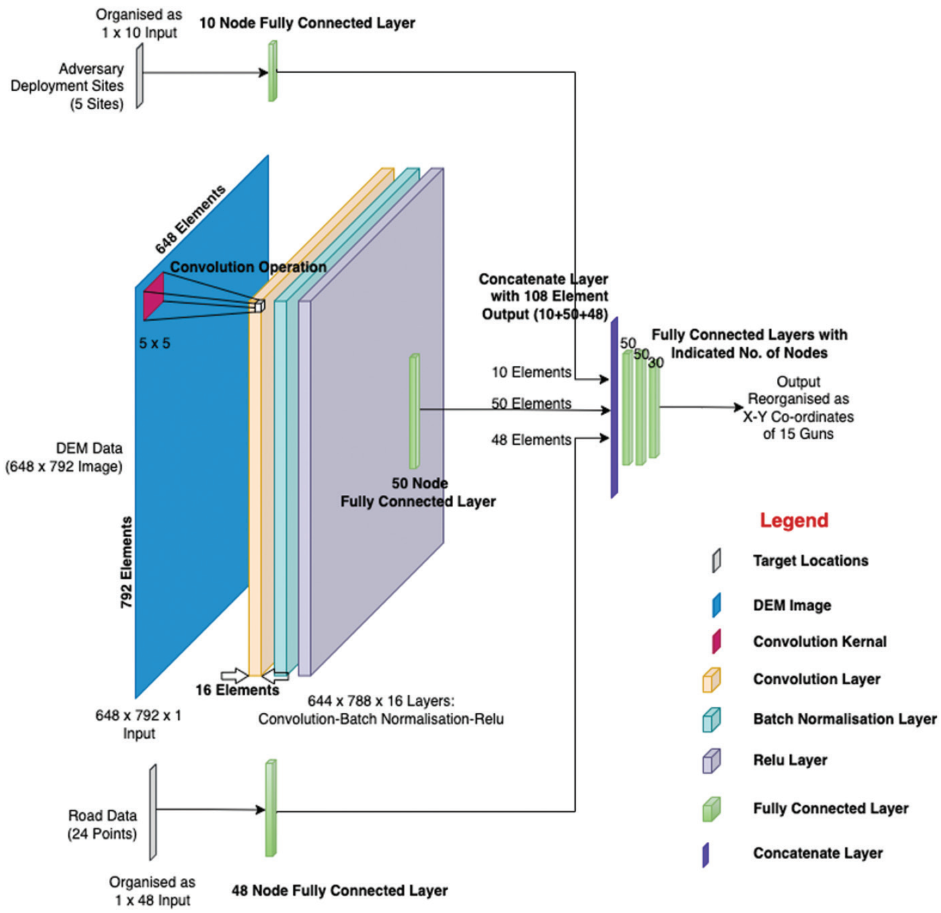


Figure 2. Neural network architecture.

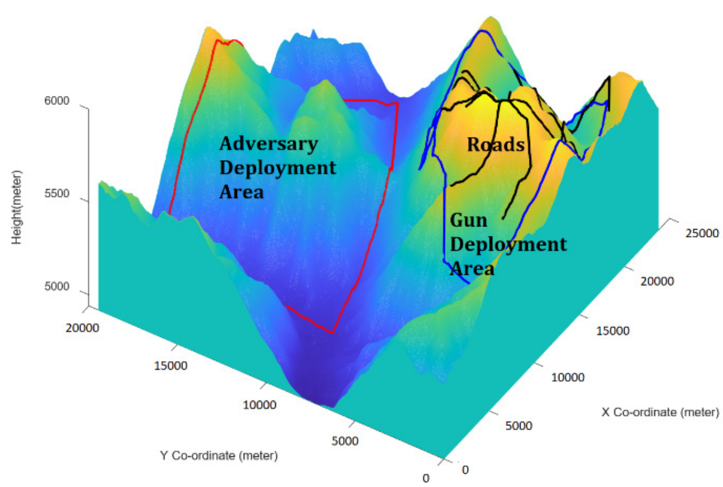


Figure 3. Scenario for dataset generation.

adversary & own force using the simulation front end (Fig. 3). User provides weights of the five factors as per the operational/tactical objective under analysis (for current purpose value of 0.1, 0.15, 0.2, 0.25 and 0.3 is assumed for O1-O5). In the current scenario, 5 adversary deployment sites are considered, corresponding to which 15 own gun sites are determined. Monte-Carlo simulation is carried out to generate the training dataset consisting of 1400 records.

Parameters used for the PP-PSO algorithm during simulation are listed in Table 3. Details of parameters can be found in our previous work[4]. When standard deviation of last n (here 15) iterations becomes lower than the threshold standard error (here, 0.0001), the algorithm is terminated.

Different datasets of varying size (200, 400, 600, 800, 1000, 1200, 1400) selected from the Monte-Carlo simulation

Table 3. PP-PSO parameter used for dataset generation

Parameter	Value
Size of swarm population	25
Inertia weight	1
Damping ratio for inertia weight	0.99
C1	2
C2	2
C4	1
Velocity scale factor	0.1

Table 4. Training data summary

Dataset size	PP-PSO MoE		Average computation time
	Mean	Standard deviation	
200	0.9129	0.0161	
400	0.9128	0.0159	
600	0.914	0.0139	
800	0.913	0.0137	183.12s$
1000	0.9137	0.0144	
1200	0.9138	0.0153	
1400	0.914	0.0149	

$ For MATLAB platform on a Windows 10 machine with 14 core Dual-Xeon processor, 64 GB RAM

Table 5. Neural Network (NN) results for test data

Dataset size	Dataset partition size		Mean MoE for test data		Standard deviation of MoE for test data		Average computation time	
	Training	Test	PP-PSO	MoP-N	PP-PSO	MoP-N	PP-PSO	MoP-N
200	160	20	0.9146	0.8232	0.017	0.0218		0.6795
400	320	40	0.9112	0.8042	0.0234	0.033		0.6761
600	480	60	0.913	0.8079	0.0197	0.0268		0.6782
800	640	80	0.9122	0.79	0.0215	0.06	183.12	0.6698
1000	800	100	0.91	**0.84**	0.02	**0.02**		**0.6731**
1200	960	120	0.9125	0.71	0.0183	0.0568		0.6621
1400	1120	140	0.9126	0.73	0.0183	0.0399		0.68

results, are used to train the network. Data is divided in 80:10:10 ratio for training, validation and testing. Training is done with a mini batch size of 80 using Adam optimizer. Mean square error is used as loss function. Value of 0.005 is used as initial learning rate. The model is trained with 100 epochs and validation performed after every 10 epochs. Network iteration with minimum validation loss is selected as the final network. Summary of training datasets is presented in Table 4.

6. RESULTS AND ANALYSIS

Results predicted by neural network (NN) with varying size of training data are presented in Table 5. Following factors are calculated to analyse the NN performance:

- Mean difference between NN MoE and PP-PSO MoE for test data
- Average computation time (for MATLAB platform on a Windows 10 machine with 14 core Dual-Xeon processor, 64 GB RAM).

The trend of MoE vs the dataset size has been plotted in Fig. 4. It depicts the comparison of test data MoE achieved through MOP-N vis-à-vis average MoE of the test data set for corresponding PP-PSO results. As the size of the dataset increases from 200 to 1000, the difference in test data MoE for MOP-N and PP-PSO decreases. When the dataset size is 1000, the average difference between MOP-N and PP-PSO MoE is minimum. As dataset size is further increased, the mean MOP-N MoE deteriorates with corresponding increase in MoE standard deviation. This indicates increase in variance of neural network output and thus overfitting of the results.

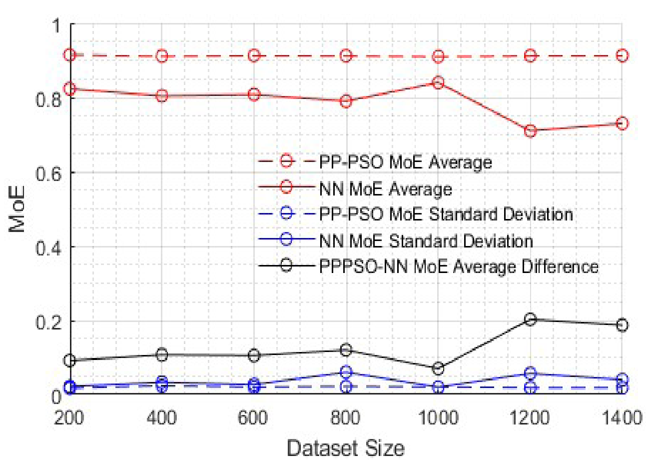


Figure 4. Comparison of MoE for PP-PSO and NN results for test data.

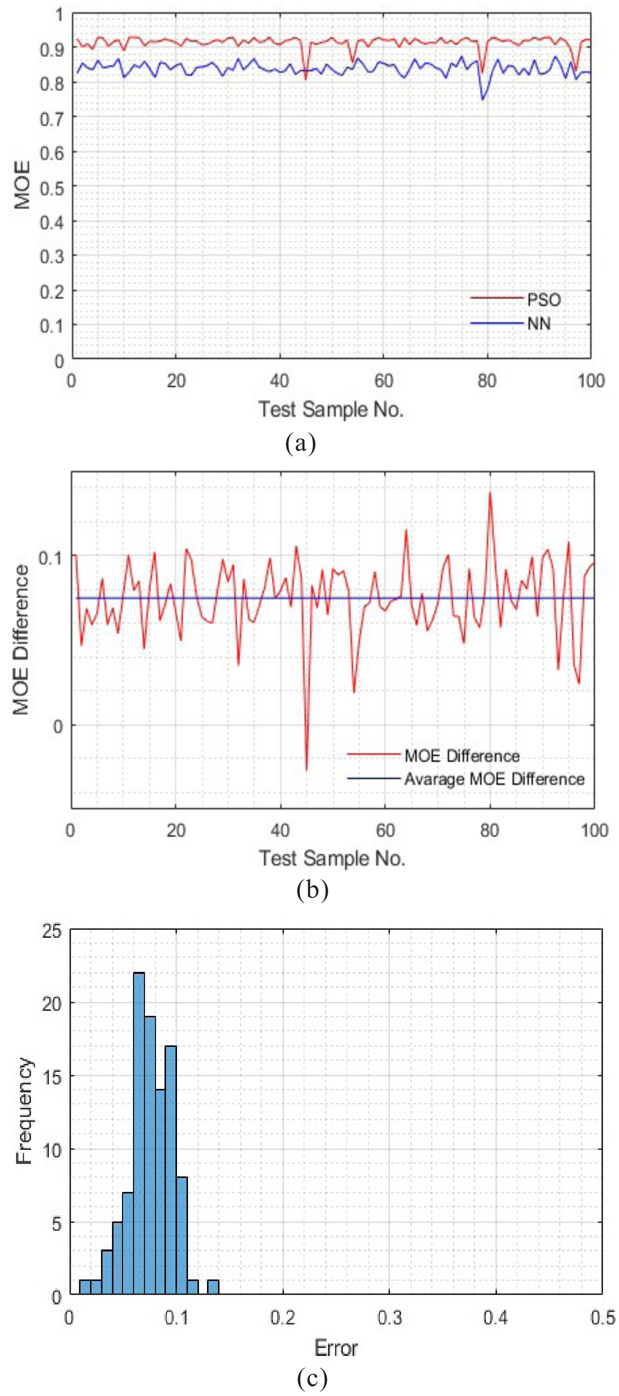


Figure 5. Plot of results for PP-PSO and NN results for dataset Size of 1000; (a) MoE (b) MoE difference for PP-PSO and NN (c) Histogram of MoE difference.

The model's generalisation capability start to decrease as the dataset size is increased beyond 1000.

Dataset size of 1000 represents the optimum choice. In this case mean the MoE is lower by a value of 0.07 w.r.t. the PP-PSO mean MoE. Each test data result for the dataset size of 1000 were further analysed through expert opinion (of participating commanders) to verify the acceptability of the results. Figure 5 shows, a comparison of results of PP-PSO and MOP-N approaches for each test data point for the dataset size of 1000. It is evident from the plots that MOP-N results usually have lower MoE than that for PP-PSO. At the same time, their values are always within the acceptable bound below the PP-PSO MoE values. As seen in Fig. 5(c), the difference is distributed around peak bar of 0.06-0.07, i.e. difference of 0.06 to 0.07 has maximum frequency of occurrence. The maximum difference has upper bound of 0.11 with the two higher values (these have frequency of 1) considered as outlier.

Figure 6 shows comparison of best-case predictions (i.e. for dataset size =1000) of neural network with that of PP-PSO algorithm. The figure indicates that the objective of near real

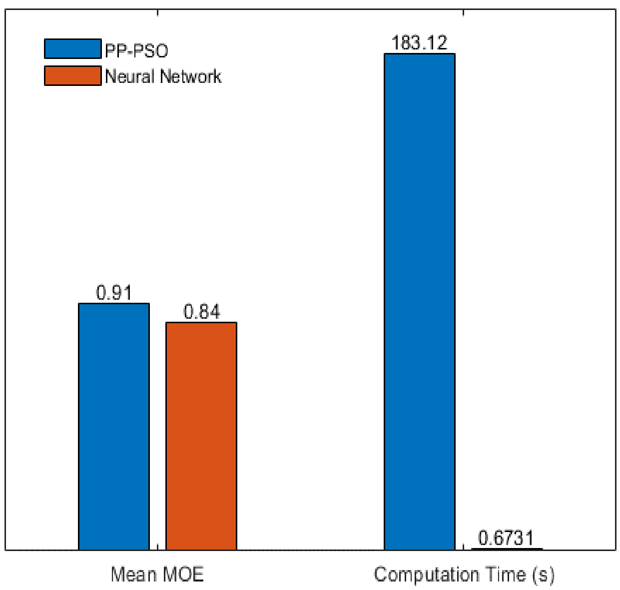


Figure 6. Comparison of neural network predictions with PP-PSO results.

time computation is achieved with use of NN as computation time for NN is less than 1 sec. with overall good mean MoE.

The neural network model resulted in encouraging results in its ability to predict the locations of own gun deployment sites in near real time for constrained dynamic scenario simulation. While by no means a perfect prediction system, a solution with achieved accuracy can serve the commanders in providing timely decision-making support in a compressed time simulation environment, where computation time is at premium.

8. CONCLUSION

Deployment site selection is one of the critical decisions that an operational commander takes as part of operational planning process. The decision is based on particular commander's tactical/operational appreciation of the scenario, it can vary from commander to commander and also with time & space for the same commander. In wargaming simulation applications, automation of commander's decision-making process so that abstract inputs can be processed at higher fidelity, is an important aspect. Deployment of gun systems in mountain terrain is one such decision-making problem and MOP-N approach presented in the paper has shown encouraging results towards solution of the problem. Neural network predicted results are close to the results computed using PP-PSO algorithm and the computations are near real time. The achieved results were within 8 % of the PP-PSO results with 99.6 % reduction in computation time.

A significant contribution of this research is in devising a hybrid approach for application of machine learning algorithms to automated military deployment problems in a data limited situation and also in establishing efficacy of machine learning algorithms for automated decision support. The proposed model demonstrates that near real time location prediction of own gun deployment sites in the dynamic environment is feasible. The deployment sites generated by the algorithm were analysed by military subject matter experts (SMEs); the locations and their dispersion were found to be satisfactory for operational use. The proposed methodology has considerably improved the computational speed while producing acceptable solutions. The solution approach may also be utilised for solving other similar problems in time efficient manner.

REFERENCES

1. Wong, Y.H.; Sebastian, J.B.; Elizabeth, M.B. & Smith, B.M. Next-generation wargaming for the U.S. marine corps: Recommended courses of action. RAND Corporation, Santa Monica, CA, 2019.
2. Bisht, S.; Taneja, S.; Jindal, V. & Bedi, P. APSO based automated planning in constructive simulation. *Def. Scie. J.,* 2023, 564-571.
   doi : 10.14429/dsj.73.5.18497.
3. Taneja, S.B.; Brahma, G.C.; Bisht, S. & Jindal, V. Predator-prey PSO based weapon deployment in constructive simulation. *In* Fifteenth International Conference on Contemporary Computing, Noida, 2023.
   doi: 10.1145/3607947.3608296.
4. Taneja, S.B.; Brahma, G.; Bisht, S.; Jindal, V. & Punam, B. Comparative analysis of PSO techniques for artillery deployment in constructive simulation. *In* 4th Congress on Intelligent Systems, Bengluru, 2023.
   doi: 10.1007/978-981-99-9040-5_25.
5. Knack, A. & Powel, R. Artificial intelligence in wargaming: an evidence-based assessment of AI applications. CETaS Research Reports, 2023.
6. Gao, C.; Song, Y.; Shen, J. & Li, Y. Solving location problem of distribution center based on hybrid particle swarm algorithm. *J. Comput.,* 2018, **13**(10), 1185-1191.
   doi: 10.17706/jcp.13.10.1185-1191.
7. Liu, L.; Li, X.; Liu, R. & Yan, J. Mathematic model of key-point anti-air position ring-deployment and optimisation. Proceedings of 2nd International Conference on Computer Science and Network Technology, 2012.
   doi: 10.1109/ICCSNT.2012.6525913.

8. Jia, Z. & Kiang, J.F. Deployment optimisation of defense stations in an attack-defense aerial war game. *Appl. Scie.,* 2022, **12**.
doi: 10.3390/app122110801.
9. Chen, C.; Jie; C. & Chunmei, Z. Deployment optimisation for air defense base on artificial potential field. *In* 8th Asian Control Conference, 2011.
10. Sun, W.; Cao, Z.; Wang, G.; Song, Y. & Guo, X. An optimized double-nested anti-missile force deployment based on the deep Kuhn–Munkres algorithm. *Mathematics,* 2022, **10**(23).
doi: 10.3390/math10234627.
11. Feng, J.; Jiang, N.; Wang, J. & Wang, J. Research on genetic algorithm theory and its application on missile fire allocation. *In* 7th World Congress on Intelligent Control and Automation, 2008.
doi: 10.1109/WCICA.2008.4594006.
12. Peng-jiao, Z. & Jian-guo, L. Deployment optimisation of air defense force deployment based on Memetic algorithm. *In* 29th Chinese Control And Decision Conference, 2017.
doi: 10.1109/CCDC.2017.7979481.
13. Freiberg, M.; McLaughlin, K.; Ningtyas, A. & Taylor, O. Enemy location prediction in naval combat using deep learning. *In* Systems and Information Engineering Design Symposium, VA, USA, 2021.
doi: 10.1109/SIEDS52267.2021.9483759.

CONTRIBUTOR

Mr S.B. Taneja obtained his MTech (Computer Science) from the IIT Roorkee, and working as Scientist 'H' at DRDO-ISSA, Delhi. He is involved in the design and development of wargame simulation systems. His areas of interest are distributed simulation, wargame simulation system design, geographical information system, and agent-based models for constructive simulation.
In the current study he has contributed to tactical domain setting, the conceptualisation of approach, model framework design, and validation.

Mr Sourabh Jaiswal obtained his MTech (RF Design & Technology) from the IIT Delhi, and working as Scientist 'F' at DRDO-ISSA, Delhi. His areas of interest are Mathematical modelling of electronic entities, operations research & systems analysis, and artificial intelligence.
In the current study he has contributed in domain setting to algorithm mapping, software implementation, testing, and result analysis.

Mr Sanjay Bisht obtained his MTech (Computer Technology and Application) from the Delhi Technological University, Delhi, and working as Scientist 'F' at DRDO-ISSA, Delhi. His areas of interest are wargame, heuristic optimisation, genetic algorithm, PSO, simulated annealing, and intelligent agent technology.
In the current study he has contributed to PSO parameters tuning, draft checking, and result validation.

Dr Vinita Jindal obtained her PhD in Computer Science from the University of Delhi and working as a Professor in the Department of Computer Science, Keshav Mahavidyalaya, University of Delhi. She is working in the area of Artificial intelligence and networks. Her areas of interest include: Covert channels and their detection, cybersecurity, intrusion detection systems, dark web, deep learning, recommender systems, swarm intelligence, and vehicular networks.
In the current study she contributed to NN architecture and parameter tuning, draft checking, result validation, and supervision & guidance.

Dr Punam Bedi obtained her PhD in Computer Science from the University of Delhi and working as a Senior Professor in the Department of Computer Science at the University of Delhi. Her research interests include: Steganography and steganalysis, cybersecurity, intrusion detection systems, recommender systems, deep learning, artificial intelligence for healthcare, and artificial intelligence for agriculture.
In the current study she has contributed to algorithm performance validation, draft checking, supervision, and overall guidance.