Adaptive Radar Resource Manager

Anindita De, Vengadarajan, D. Shashikiran and Bibhabasu Modal

***Abstract*—Multi-function phased array radars carry out a large number of functions like surveillance, confirmation of search detections, tracking, weapon guidance, kill assessment to list a few. However, radar resources like time and energy required to execute these functions are limited. So an optimum allocation of resources to ensure good performance in these radars is a challenging problem. Radar resource manager (RRM) carries out this job in radars. Each function is sub-divided into tasks and each task is further divided into looks. RRM interleaves looks from different tasks to execute the functions. Typically, the tasks are prioritized and during insufficient availability of resources the high priority tasks are serviced and the low priority ones are dropped. This ensures graceful degradation in the performance of the radar under overload. Hence task prioritization is crucial for an efficient RRM. Most of the operational radars use fixed priority schemes that are decided prior to the operation of the radar. Such techniques are sub-optimal in dynamic target and clutter scenarios. Improvements are possible with adaptive RRM algorithms. In this paper, we analyze the scope of neural networks for adaptive task prioritization. The technique is promising and with the advent of computers with high processing power there exist scope for practical implementation of the technique in future radars.**

***Index Terms*—neural networks, radar resource manager (RRM), task prioritization, radar schedulers**

I. INTRODUCTION

Traditionally, dedicated radars were used to carry out different functions like surveillance, tracking and weapon control. However, with the advent of multi-function phased array radars, today it is possible to carry out all these functions using a single radar. This becomes possible by actively controlling the different parameters like beam position, dwell time, waveform, and energy of the radar. The role of the Radar Resource Manager (RRM) is to coordinate different radar functions and ensure optimal performance. Hence, RRM is very critical to the success of the radar.

Each radar function is split into different tasks and each task is further sub-divided into looks. Look is a continuous time interval that needs to be completed without a pause. RRM considers interleaving of looks from different tasks for an efficient radar.

There are three major radar resources: time, energy, and processing power. To complete each task certain amount of these resources are required. Time is defined by the tactical requirements, energy is defined by the transmitter energy and processing power is constrained by the RRM computer. Time resource is the most critical of all the three since it is not possible to create additional radar timeline. Processing budget is the least constraining of all with the new generation computers. Different loading conditions faced by the radar are: underload, nominal load, and overload. The role of RRM becomes more critical when the resources are insufficient for all the tasks i.e. during overload. Under such circumstances the tasks are prioritized and the high priority tasks are serviced while the lower priority tasks are delayed or dropped. [1] proposes a generic model of RRM which is summarized in Fig.1. The mission profile of the radar determines which functions are to be carried out. Based on that a list of radar tasks is prepared and given as input to RRM. The RRM algorithms ensure optimum allocation of resources to tasks after prioritization and a task schedule is generated as output. This translates to a new beam being played by the radar with suitable waveform in the sequence suggested by RRM resulting in detections from the environment (targets and clutter). Based on this the next set of radar functions are decided and the loop continues.

Existing radars consider fixed task set with fixed priorities. E.g. Radar functions like weapon control, tracking and surveillance are assigned priorities in the decreasing order before operating the radar. Under overload conditions surveillance tasks will be delayed or dropped. However pre-fixed priority assignments have shortcomings in dynamically changing radar environment. Hence the need arises for adaptive RRM algorithms. In addition to this current and future radar systems should also allow all the radar functions to be carried out through software controls. Potential benefits of adaptive techniques include optimum use of the radar timeline with improved performance when close to overload, ability to rapidly reconfigure the radar to suit the dynamic environment, adaptive modifications to radars' performance as the environment changes.

In this paper, we focus on adaptive prioritization of radar tasks with dynamic environment conditions. We explore in detail the scope of neural networks for task prioritization and its potential for building an adaptive RRM.

This paper is divided into five sections. Following a brief introduction to Radar Resource Manager in section I, we give an overview of various RRM algorithms available in literature in section II. Section III brings out theoretical background of adaptive task prioritization scheme based on Neural Networks followed by simulation studies and results in sections IV. Conclusions are presented in section V.

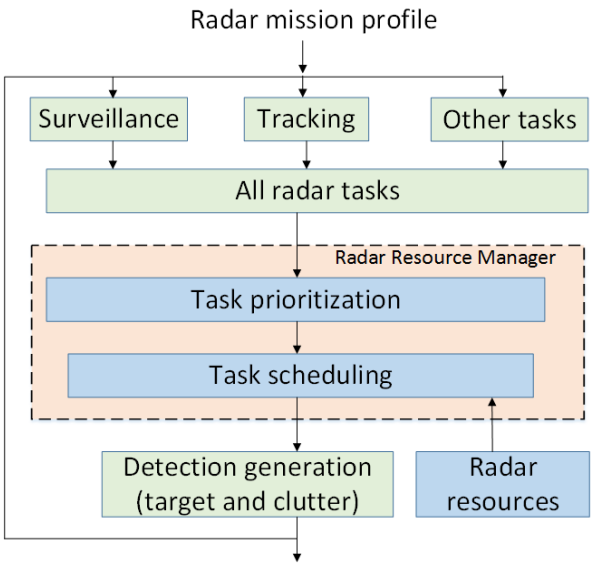

Fig. 1. Radar Resource Management model

II. Overview of RRM Algorithms

Broadly the RRM algorithms studied in literature are summarized by [9] is captured in Fig.2.

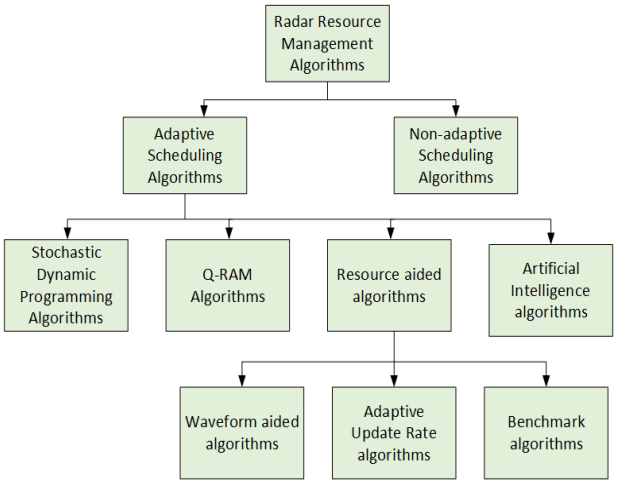

Fig. 2. Overview of RRM algorithms

The non-adaptive algorithms are used by heuristic schedulers which have rule based design. The behavior of both prioritization and scheduler modules are pre-defined by rules which are fixed. In contrast, the adaptive algorithms dynamically determine prioritization and scheduling to optimize the overall performance. An overview of adaptive techniques suggested in [1],…[9] is presented in the following sub-sections.

A. *Artificial Intelligence Algorithms*

These algorithms can be used for task prioritization and scheduling. Different approaches possible in this category are based on techniques like neural networks, expert systems and fuzzy logic. Neural networks are trained offline using appropriate datasets and can be used in operational scenarios later. Expert systems consist of knowledge database based on heuristics and an inference engine that acts based on this database. Fuzzy logic is used to assign target priorities based on features with vague values like e.g. friendly or dangerous. Such techniques are efficient if the prior knowledge captured in the form of the database is exhaustive.

B. *Optimization Theory based Algorithms*

Dynamic programming based algorithms formulate the task prioritization and task scheduling problems into one cost function. The technique is optimization theory based and the decisions are made in stages. Such algorithms are computationally complex. However, the increase in processing power has made these algorithms more practical.

C. *Resource Aided Algorithms*

The performance of the radar can be improved by modifying various parameters. E.g. A sub-class of this category i.e. waveform aided algorithms provide a noticeable improvement when there are jamming resources. Radar detection performance is improved by choice of optimum waveform. The waveform parameters are chosen based on changing environmental conditions like eclipsing, clutter, propagation and jamming. Another sub-class is adaptive update rate algorithms. This group of algorithms aim at improving the tracking performance of radars which traditionally would use uniform update rates for clutter and maneuvering targets.

III. Adaptive Task Prioritization

The need of task prioritization arises when the total time available for scheduling is less than the total duration of tasks to be scheduled. So an efficient prioritization scheme will ensure that the radar's performance is optimal even in dynamically arising overload conditions. Present operational radars use fixed prioritization schemes which give sub-optimal performance in such scenarios. Hence the need for adaptive task prioritization arises.

Although the schedulable entity is task, we can aim at learning priority values or ranks of detected targets or objects after threat assessment as suggested in [2]. The tasks which resulted in these detections can inherit the ranks from the respective detections. It is assumed that every task is connected to a certain object that should be scheduled and executed. The mapping between detected objects and their corresponding tasks is described in Fig.3. E.g. Detected target T_1 with priority P_1 arises due to execution of two tasks. Under operational conditions tasks are ranked according to the learned models.

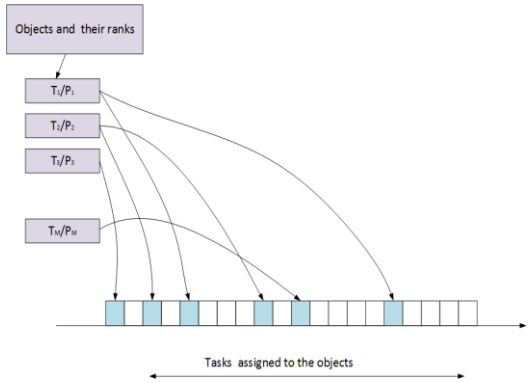

Fig. 3. Mapping of detected targets (T_1, … T_M) to tasks that resulted in these detections

In this paper, we focus on neural network based prioritization. There are two parts in this technique: training and validation followed by testing. Training set considered is a set of N samples, where each sample is a pair of feature vector and priority. As suggested in [3] feature vector X suitable for the problem is formulated as follows:

$$X = (x_1\ x_2\ x_3\ x_4\ x_5\ x_6)'$$

where x_1 : range of the target (in km)
x_2 : radial velocity of the target (in m/s)
x_3 : signal: friend (x_3 =0), foe (x_3 =1)
x_4 : acceleration of object (in m/s^2)
x_5 : object rank ($x_5 > 0.5$ for important targets, $x_5 < 0.5$ for less important targets)
x_6 : direction of motion of the object (x_6 =0 for objects moving towards the radar, x_6 =1 for objects moving away from the radar)

The training set is defined by $U =< X^i, z^i >_{i=1}^N$ where X is the feature vector z is the corresponding priority.

The input to the neural network is the feature vector and the output is priority value/rank of the detected objects. Any complex mapping of the input to the output can be learned using neural networks. The network is characterized by a number of layers between the input and the output. The length of the feature vector defines the number of nodes in the input layer. Similarly, the number of nodes in the output layer is determined by the number of outputs. The layers in between are called hidden layers. The degree of freedom of the network is defined by the number of hidden layers and also the number of nodes in each hidden layer. A neural network in its simplest form is shown in Fig. 4.

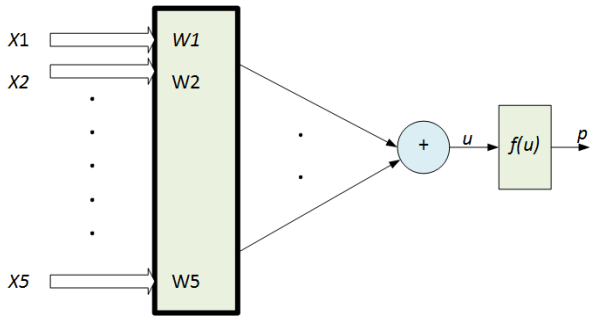

Fig. 4. Neural network in simplest form

A weighted sum of the feature vector gives the input stimulation signal, u which is passed through an activation function f. In this paper, sigmoid function is used for activation. The relevant equations are given below:

$$u = \sum_{i=0}^{n} w_i\, x_i \quad (1)$$

$$f(u) = \frac{1}{1+e^{-bu}} \quad (2)$$

The slope of the activation function depends on parameter b. The weights are learned by back-propagation which involves minimizing the mean square error. It can be defined as

$$k = \frac{1}{2} \sum_{i=0}^{n} \alpha_i^2 \quad (3)$$

where $\alpha_i = z_i - p_i$, is the difference between the requested rank, z_i and the computed rank after activation p_i of sample number i. The weights are calculated by minimizing the error k over the chosen learning set U. The learned model generalizes the target rank and assigns priorities to unseen targets during the testing phase. The generalization error is kept minimum by selection of an appropriately sized dataset for learning.

IV. EXPERIMENTS

A. *Training set generation*

The dataset used for learning can be generated using simulated or real target data. In this paper, a training set of 2100 samples is simulated using MATLAB. The setup used is as given in Fig. 5.

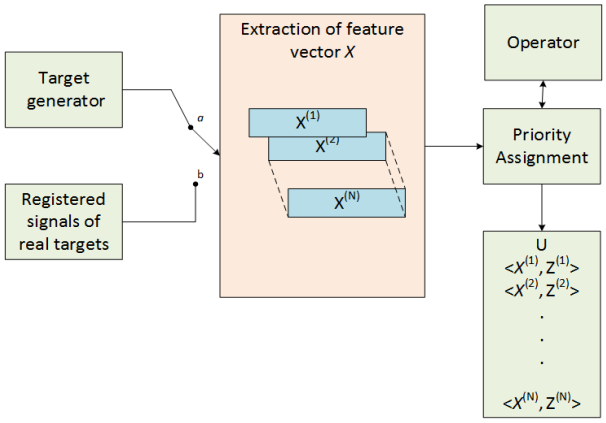

Fig. 5. Training set generation module

Trajectories of different types of targets like missiles, ships and aircrafts are simulated using the target generator module. Equal number of samples are simulated for each target type. The velocity and acceleration of the targets are chosen appropriately to suit the target type. All the enemy targets are assumed to be approaching the radar while the friendly targets are randomly chosen to be approaching or receding from the radar. All the enemy targets are considered important while the friendly targets are considered important if they have a high velocity and acceleration else less important. The degree of importance is incorporated in the relevant feature during the formatting step. Prior knowledge of the operator is used to define the priorities of each training sample. In this paper, a total of six priority

TABLE I
PRIORITY LEVELS CONSIDERED IN THE TRAINING SET

Target	Feature	Priority level
Missiles	Foe	1
Aircrafts	Foe	2
Ships	Foe	3
Missiles	Friendly, high velocity	3
Missiles	Friendly, low velocity	4
Aircrafts	Friendly, high velocity	4
Aircrafts	Friendly, low velocity	5
Ships	Friendly, high velocity	5
Ships	Friendly, low velocity	6

levels are simulated. Different cases of priority levels are summarized in the Table I.

B. Results

The simulated dataset is divided into three subsets: 70% of the samples are randomly picked up for training, 15% for validation and rest 15% are used for testing. The number of hidden layers used are eight.

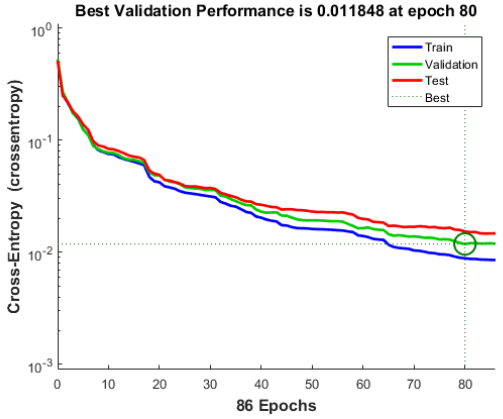

Fig. 6. Performance variation with epochs

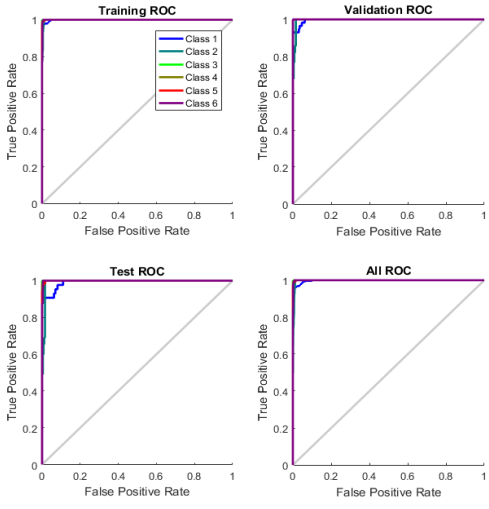

Fig. 7. ROC curves for training, validation and testing

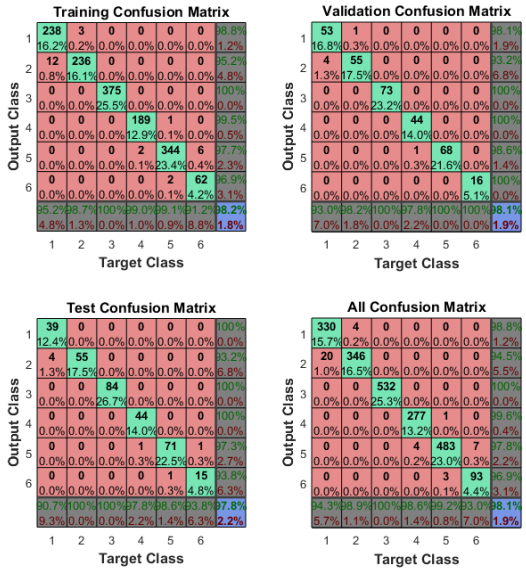

Fig. 8. Confusion matrix for training, validation and testing

V. CONCLUSION

Neural networks have good scope in the area of task prioritization. They can learn any complex relation between the input and the output. This makes them very useful in learning abstract relation between task features and priorities. However, their success depends on how well the learning dataset is designed. The size of the learning set is crucial too, as an extremely big dataset will result in poor generalization due to overfitting. We aim to explore this technique with real data in future.

REFERENCES

[1] Peter W Moo and Zhen Ding, “Adaptive Resource Management,” in *Elsevier*
[2] W. Komorniczak, J. Pietrasinski and B. Solaiman, "The data fusion approach to the priority assignment in the multifunction radar," 14th International Conference on Microwaves, Radar and Wireless Communications. MIKON - 2002. Conference Proceedings (IEEE Cat.No.02EX562), 2002, pp. 647-650 vol.2.
[3] W. Komorniczak, Tomacz Kusczerski and J. Pietrasinki, “The priority assignment for detected targets in multi-function radar”.
[4] S. L. C. Miranda, C. J. Baker, K. Woodbridge and H. D. Griffiths, "Fuzzy logic approach for prioritisation of radar tasks and sectors of surveillance in multifunction radar," in IET Radar, Sonar & Navigation, vol. 1, no. 2, pp. 131-141, April 2007
[5] M. T. Vine, "Fuzzy logic in radar resource management," IEE Multifunction Radar and Sonar Sensor Management Techniques (Ref. No. 2001/173), 2001, pp. 5/1-5/4. doi: 10.1049/ic:20010183
[6] S. Miranda, C. Baker, K. Woodbridge and H. Griffiths, "Knowledge-based resource management for multifunction radar: a look at scheduling and task prioritization," in IEEE Signal Processing Magazine, vol. 23, no. 1, pp. 66-76, Jan. 2006.
[7] S. L. C. Miranda, C. J. Baker, K. Woodbridge and H. D. Griffiths, "Phased array radar resource management: a comparison of scheduling algorithms," Proceedings of the 2004 IEEE Radar Conference (IEEE Cat. No.04CH37509), 2004, pp. 79-84.
[8] W. Komorniczak and J. Pietrasinski, "Selected problems of MFR resources management," Proceedings of the Third International Conference on Information Fusion, Paris, France, 2000, pp. WEC1/3-WEC1/8 vol
[9] Radar Resource Management technique for multi-function phased array radars, PhD thesis, Omer Cayir

Anindita De was born in Calcutta, India in 1987. She received B.Tech in Electrical and Electronics Engineering from National Institute of Technology, Calicut in 2009 and M.E in System Science and Automation from IISc, Bangalore in 2016. She has worked in the area of radar signal and data processing for phased array radars. Her research interests include sub-Nyquist sampling and Machine Learning.

D. Shashikiran was born in Bangalore, India in 1980. He received B.E in Electronics and Communications Engineering from the National Institute of Technology, Suratkal in 2002 and M. Tech in Telecommunications System Engineering from IIT Kharagpur in 2010. He has worked in the realization of C4I systems and Radar Systems. His areas of interest are Radar signal processing and Radar system engineering.

Bibhabasu Mondal was born in 1983. He has obtained his B.Tech degree in Electronics and communication Engineering in 2005 from West Bengal University of Technology,West Bengal.