https://doi.org/10.1590/jatm.v13.1228 ORIGINAL PAPER

Decision Support System for Unmanned Combat Air Vehicle in Beyond Visual Range Air Combat Based on Artificial Neural Networks

Geraldo Mulato de Lima Filho[1,*], Felipe Leonardo Lôbo Medeiros[2], Angelo Passaro[1,2]

1. Departamento de Ciência e Tecnologia Aeroespacial – Instituto Tecnológico de Aeronáutica – Programa de Pós-Graduação em Ciência e Tecnologias Espaciais – São José dos Campos/SP – Brazil. 2. Departamento de Ciência e Tecnologia Aeroespacial – Instituto de Estudos Avançados – Programa de Pós-Graduação em Ciência e Tecnologias Espaciais – São José dos Campos/SP – Brazil.

*Correspondence author: geraldolfi@hotmail.com

ABSTRACT

In a beyond visual range (BVR) air combat, one of the challenges is identifying the best time to launch a missile, which is a decision that must be made quickly. The decision involves combining knowledge about altitude, speed, distance, onboard sensor systems information, aircraft type, and type of missile on the aircraft, as well as intelligence on the opponent's behavior. This paper discusses an approach to evaluate the probability of shoot-down of an unmanned combat air vehicle (UCAV) in a BVR air combat, based on a decision support system model that makes use of parameters available from the onboard sensors of the shooter UCAV. The strategic options development and analysis (SODA) method is applied to select the main features available in the on-board sensor systems of the shooter aircraft required to launch a missile successfully. Such features help us to develop an artificial neural network (ANN) for shoot-down prediction. The ANN was trained with a data set with 1093 registered shoots in military exercises, and it shows 78.0% accuracy with the cross-validation procedure.

Keywords: Machine learning; multilayer perceptron; Strategic Options Development and Analysis.

INTRODUCTION

Beyond visual range (BVR) air combat has become the most important type of contemporary air combat. Beyond visual range combat occurs with the detection and tracking of an enemy aircraft (target) beyond the range of the pilot's vision (Liang *et al*. 2010). Note that the pilot's vision limit depends on the existence of visual obstacles (clouds, smoke), visual acuity, devices to improve visual acuity, luminosity, aspect and size of the target (Galante 2010). Beyond visual range combat uses radar sensors and radar warning receivers (RWR) to detect and track targets and specific air-to-air missiles.

Beyond visual range combat involves a highly complex operating environment and requires cognitive knowledge, skills and a huge mental effort on the part of the pilot. A complex environment generates cognitive workloads, degrading operator performance.

Received: Jan. 19, 2021 | **Accepted:** Ago. 2, 2021
Peer Review History: Single Blind Peer Review.
Section Editor: Ana Morais

Gilberto (2017) states that failures can be summarized as pilot-caused operational errors of the aircraft weapon system when the missile is launched. Errors from the aircraft sensors, such as radar errors in estimating the position and speed of a target, can also degrade the application of the aircraft weapon system. These errors lead to the loss of armament efficiency in reaching the target. Several other factors involved in this environment can change the outcome of the mission. Such environmental factors resulted in the intensification of the use of unmanned combat air vehicle (UCAV) and artificial intelligence (AI) in BVR scenarios in the last decade.

The main phase of BVR combat is engagement. In a simplified way, this phase is a sequence of the following activities: the UCAV performs the target tracking procedure; decides whether to fire, checking the possibility of fire; if so, carries out the triggering procedure; performs the trigger support procedure, and makes an evasive maneuver, ending the engagement.

Identifying the right shoot time and the probability of success of the shoot is important for BVR combat to achieve the high strategic value of the mission. The shooter would like to increase the probability of success of each missile launched. In traditional BVR combat, the pilot launches the missile inside the weapon engagement zone (WEZ), a dynamic weapon range area calculated using data from the onboard sensors about the missile range, target and shooter's altitude, target and shooter's speed, and heading difference between the target and the shooter (Macedo 2017). The pilot does not know the probability of success at launch time because success probability combines several elements beyond those used in WEZ, such as target maneuvers, shoot range, number of aircraft in formation and closure rate.

Several publications in the literature address different aspects of BVR combat, from the allocation of targets to the analysis of the armed clash between opposing forces. Various methods are adopted, from knowledge bases to elaborated mathematical models aimed at computer simulations as a support for decision-making.

Rodin *et al.* (1987) proposed using an expert system in conjunction with pursuit-evasion algorithms and a pilot-derived expert database to evaluate alternative strategies for air combat. Rodin and Amin (1992) formulated and solved problems that involve prediction and identification of continuous trajectory air combat maneuvers using partial or incomplete information. Brain (1999) did a conceptual study using an artificial neural network (ANN) as a decision-support system for combat- maneuver-identification into BVR combat environments.

More recently, Yu and Wu (2011) addressed a methodology that uses fast arithmetic to evaluate the air target threat, helping pilot make decisions in a dynamic combat environment. Wu *et al.* (2013) consider the uncertainty of air combat information to threat assessment based on rough sets and support-vector machines (SVM). The tactics decision problem for cooperative team air combat is the focus of Meng *et al.* (2014), in which a granular computing theory is applied to the tactics decision analysis. Suseno and Sasongko (2016) use an air combat simulation system, which includes fighter maneuver modeling, definition, and formulation of parameters and probability of kill, to assess the effectiveness of maneuvers, tactics, and weapons in a combat situation.

Toubman *et al.* (2016) developed a machine learning method that can rapidly adapt the behavior of computer-generated forces (CGFs) to that of their opponents. The method entails the adaptation of behavior represented as finite-state machines (FSMs) through a reinforcement learning technique called dynamic scripting. Lei *et al.* (2018) use a machine learning approach with data samples generated by a "human-in-loop" air combat simulation system. The data includes the aircraft model, attributes of the two opposing groups (blue and red), aircraft position, attitude, and critical actions (target locking, real-time interference, launch missiles). Ximeng *et al.* (2019) also adopt a machine learning approach to target threat assessment. They use one-to-one air combat data from the air combat maneuvering instrument (ACMI), such as the relative position of the aircrafts involved, speed vectors, and relative angles, as input for the neural network.

Ma *et al.* (2019) address the problem of optimization of the predominance between two opposing groups of UAVs in a BVR combat scenario. One UAV group predominance over the other is estimated by the difference in distance and altitude between them, the lower and upper limits of the missile shooting distance, and the best shooting altitude. The decision variables are the spatial distribution of the two groups and the allocation of targets for each aircraft in these groups. The optimization problem is modeled as a zero-sum game and is solved in order to obtain a Nash equilibrium.

Wanyang *et al*. (2020) focus their research on the advantages of the situation and the effectiveness of air combat, taking into account in their mathematical modeling the advantage of the geometric situation (angle advantage, distance advantage, speed advantage) and the effectiveness of combat training (lead launch probability, kill probability of missile to target, air combat effectiveness calculation model). Finally, the Hungarian algorithm target assignment method is used to study the decision-making problem of cooperative multi-target attack air combat, and the ideal formation attack assignment decision-making scheme is obtained.

The cited papers have mainly provided decision support that improves tactics and strategies in air combat and the target allocation problem. The few studies addressing the probability of an aircraft being shot down by a missile use computer simulation based on mathematical models. No publications estimating the probability of a missile firing success using actual fight data obtained from military exercises and ANN were found. This type of problem has characteristics such as many input variables, noise, and the need for real-time processing, conducive to treatment by an ANN. Unmanned combat air vehicles must optimize the use of their weaponry by evaluating the probability of success before the launch of their missile, improving the efficiency regarding the human being that relies on WEZ, on its experience, and its response time. The main contribution of this paper is the use of information available in the aircraft (onboard sensor systems) collected in military exercises at the moment of missile launching to develop an ANN for predicting the shoot down of the opponent. The strategic options development and analysis (SODA) method is applied for structuring the problem. The target is supposed to be assigned at a previous time of the engagement, and the issue is not addressed here.

In summary, the purpose of the study is to develop a decision support system based on ANN that collects the vehicle and opponent relevant data, both provided by the aircraft sensors and decides in real-time whether the pilot or UCAV will shoot or not.

The two methods used in this work (SODA and ANN) are well known in the literature (Eden and Ackermann 2001; Russell and Norvig 2016). Despite that, the basic concepts necessary to understand the methodology and the tests carried out are presented in the following sections.

This paper is organized as follows: first, it is presented the method of structuring the problem used for predicting the accuracy of a missile launch in BVR combat. Second, it is presented the general concepts of a multilayer perceptron. The following presents the methodology used to customize a neural network to be applied to the work in question. The next section it is showed the results and analysis. Finally, the last section summarizes the conclusion of this work.

STRUCTURING THE PROBLEM

At the time of launching the missile, the pilot is faced with a multicriteria decision analysis (MCDA) problem, because he must consider several criteria before shooting. Structuring problems for MCDA have been attracting increasing attention over the past 20 years, and combining problem structuring approaches with MCDA produces a richer view of the decision situation. However, there are some limitations and challenges in combining both, most of which relate to building a value tree and assigning weights for each criterion (Marttunen *et al*. 2017).

Artificial neural network is a powerful option to build a value tree. In an ANN, the weight of each node is updated using some learning rule (Mingoti and Lima 2006). At the end of the machine learning process, the weights naturally produce a value tree, as well as an optimized weight assignment. The main features of the problem, also called the decision variables, must be identified to build an ANN. In this study, these factors influence the shoot down in BVR combat. The decision variables are the input nodes in the first layer. Data from military exercises of BVR combat collected over five years were used. From this database, the data from 1489 shoots associated with the engagement between two aircraft of the same type armed with the same missile were selected. However, these data include various highly complex BVR combat arenas with a varied number of aircraft in both the attacking squadron and the target squadron, at different levels of altitude, speed and heading. Only data available to the shooter aircraft are used.

Strategic options development and analysis

Strategic options development and analysis is a problem-structuring approach that enables problematic situations to be explored in greater depth by constructing graphical representations before making a decision. Strategic options development and analysis uses a technique called cognitive or causal mapping, which analyses interviews with stakeholders and examines relevant documents. The picture is constructed using the natural language of the problem owners and becomes a model of the situation "owned" by the people who define the problem (Eden and Ackermann 2001).

Strategic options development and analysis has been used successfully for a wide range of purposes and applications. In this case, it is used to structure thinking through capturing chains of procedures and desirable pilot behaviors in the missile launch time in a BVR combat. According to Ackermann and Eden (2010), SODA allows capturing all the statements (constructs) along with their relationship (causal relationship) maps, identifying the key issues, evaluating the breadth of considerations, and identifying inconsistencies in arguments.

Georgiou (2011) says that language consists of words, many of which can have multiple connotations. One manner of minimizing the breadth of interpretation is to offer an alternative word or phrase that can highlight what is meant by the original description. This highlighting is the function of constructs, which are a concept or theoretical construction: purely mental, elaborated or synthesized based on simple data, from observable phenomena, which help researchers to analyze and understand some aspect of a study or science. Constructs are designed with two poles, whereby the second pole serves to clarify what is meant by the first pole. The two poles are separated by suspension points, distinguishing the two poles of the construct (Georgiou 2011).

Georgiou (2011) states that constructs may be structurally categorized according to certain basic types: tails, heads, strategic options, implosions, explosions and dominants. Tails have no constructs leading into them and are known as prime causes. Heads have no constructs leading out of them and reflect objectives, outcomes, results or consequences stemming from the dependency paths of arrows that lead into them. Strategic options are constructs with immediate links to a head and reflect the options available through which a particular result (head) may materialize. Implosions are constructs with a relatively high number of constructs leading directly into them; an implosion is a construct affected by multiple other constructs. Explosions are constructs with a relatively high number of constructs directly leading out of them; an explosion is a construct that affects multiple other constructs. Dominants are constructs with a relatively high total number of constructs leading into and out of them; dominants will affect, and be affected by, multiple constructs, and offer a good indication of the major issues that must be tackled in order to reach the heads. These concepts are discussed in more detail with the example below.

Application of the SODA method

In qualitative research, some techniques are used to organize data in terms of their properties and dimensions. When no new information emerges during coding, i.e., when no new properties, dimensions, conditions, actions, interactions, or consequences are evidenced in the data, a category is considered saturated (Strauss and Corbin 1998). In this study, the interview of five specialists from different origins was enough to gather the relevant information during the SODA method utilization.

Five pilots from different squadrons who participated in the BVR combat exercises were interviewed. The BVR combat manuals were consulted to map the factors that influenced the combat results. After collecting the pilots' responses and creating a cluster of causes and effects from each interview, the questionnaire answers and the individual clusters were presented to the pilots to create a consensual unified version. The pilots worked together to create a unified questionnaire response (Table 1) and a single cluster (Fig. 1) that removed the ambiguity from technical terms and standardized the cause-and-effect relationships.

It is possible to identify the main constructs in Fig. 1 according to their basic types: tails (constructs 1 to 6), heads (construct 14 in an oval), strategic options (constructs 12 and 13), implosions (constructs 12 and 13), explosions (constructs 9 and 11), and dominants (construct 9).

The information presented in the cluster was the starting point to select the desirable features in the ANN. The data needed to assess the probability of success are those available from the onboard sensor systems: difference in altitude between opposing aircraft, shooting distance, shooting position concerning WEZ (short, medium, or long shoot), minimum shooting distance, closure rate, shooter aircraft speed, angle of the opposing aircraft in relation to the shooter aircraft and the number of planes in the shooter's and the opponent's squadrons.

Table 1. Questionnaire provided to the pilots and their unified answers.

Question	Responses
What are the factors that can increase the probability of success of shooting down the opponent in the missile launch time?	The aircraft approaches the opponent.
	Aircraft as fast and as high as possible.
	Long/medium launch.
	The aircraft's aspect in launch time is favorable.
	Short launch.
	The number of aircraft in the squadron is higher than the opponent.
Why?	The closure rate is high.
	The missile can reach a longer distance.
	The opponent aircraft did not do the defensive maneuver.
	The opponent aircraft do not have enough time to escape.
	The higher the number of aircraft, the higher is the probability of shooting down the opponent.
How effective is this factor?	The missile has a low/medium probability of hitting the target.
	The missile has a high probability of hitting the target.

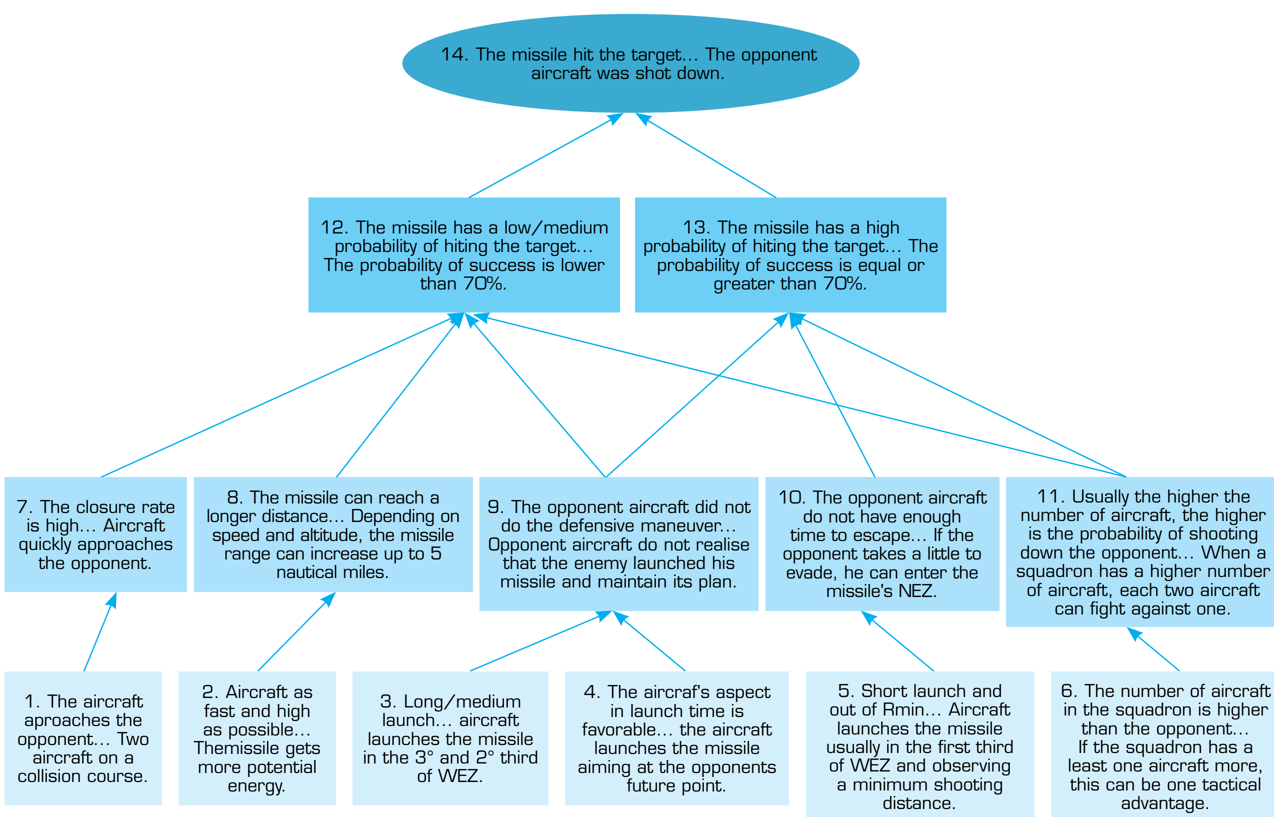

Figure 1. Unified cluster — influencing factors in shooting success.

CONCEPTS OF MULTILAYER PERCEPTRON (MLP)

Machine learning aims to program computers to use example data or past experience to solve a given problem (Alpaydin 2014). Supervised learning is learning from a training set of labelled examples provided by a knowledgeable external supervisor (Sutton and Barto 2018). In supervised learning, the teacher (or external supervisor) has knowledge of the environment, with the knowledge being represented by a set of input-output samples, called the training dataset, and the objective of learning is to estimate the free parameters of the network that minimize the output errors (Suresh *et al.* 2013). Neural networks are a type of supervised learning network in which signals are transmitted from input nodes to output nodes.

Russell and Norvig (2016) state that a neural network is composed of nodes or units connected by directed links. Figure 2 shows a representation of a single-layer neural feed-forward network, or perceptron network. A link from unit *i* to unit *j* serves to propagate the activation from *i* to *j*. Each connection also has a numerical weight associated with it, which determines the strength and the connection signal. Each layer has a bias unit input $a_0(x_0) = 1$ with associated weight $\Theta_{j,0}$. The mathematical model for an individual "unit" (also named "neuron") is summarized in Eq. 1. Each unit *j* first calculates a weighted sum of its inputs of the layer l and then applies an activation function *g* (called perceptron) to the sum to get the output:

$$a_j^{l+1} = g\left(\sum_{i=0}^{n} \Theta_{j,i}^{l} a_i\right) \tag{1}$$

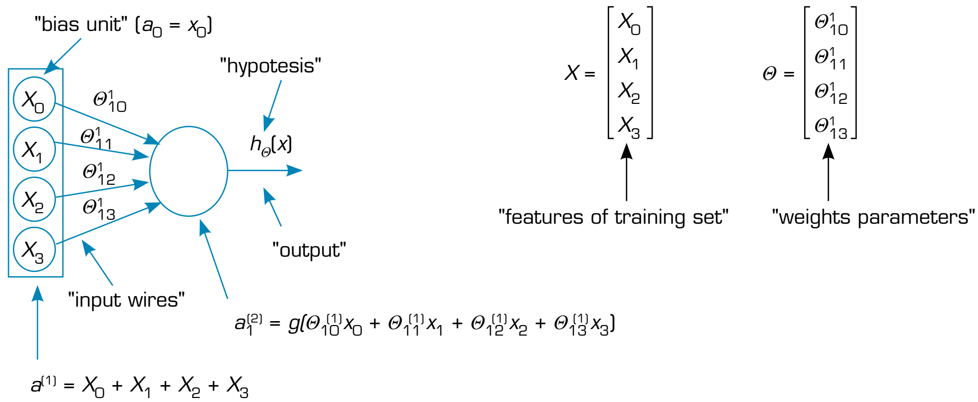

Figure 2. A simple four-input, one output perceptron network.

The next task is to connect the units to form a network. A feed-forward network has connections in only one direction (Fig. 3), forming a directed acyclic graph. Every node receives input from "upstream" nodes and delivers output to "downstream" nodes; there are no loops (Russell and Norvig 2016).

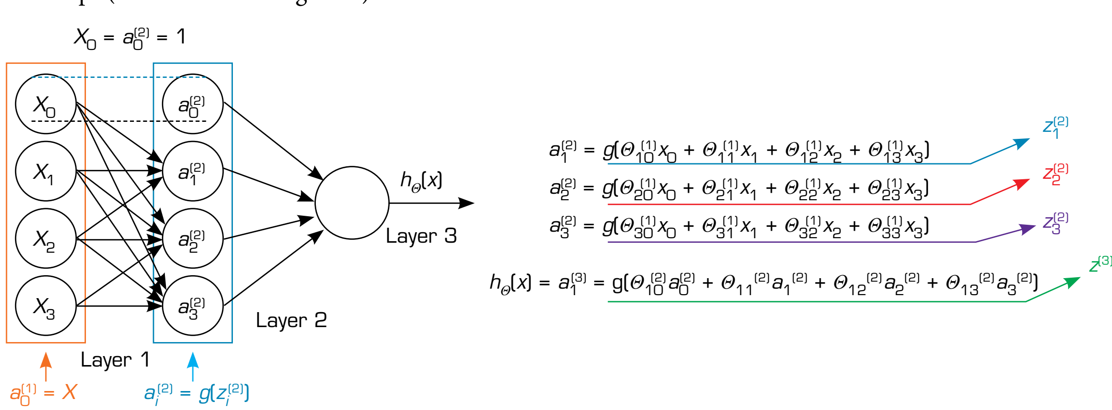

Figure 3. Multilayer perceptron with three layers (Layer 1: three input units, Layer 2: one hidden layer of three units, and Layer 3: one output unit).

A multilayer perceptron (MLP), also often called a feed-forward neural network (Goodfellow *et al.* 2016), is a supervised learning algorithm that learns a function $f(.){:}R^m \rightarrow R^0$ by training on a dataset, where m is the number of dimensions of input and o is the number of dimensions of output. Given a set of features $X_i = \{x_i|x_1,x_2,...,x_m\}$ and a target y, an MLP can learn a nonlinear function approximator for either classification or regression. Between the input and the output layer, there can be one or more nonlinear layers, called hidden layers. The first layer, known as the input layer, consists of a set of units $X_i = \{x_i|x_1,x_2,...,x_m\}$ representing the input features. Each unit in the hidden layer transforms the values from the previous layer with the application of an activation function $g(.){:}R^m \rightarrow R$ on the weighted linear summation $\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + ... + \Theta_{1m}^{(1)} x_m$. The result feeds the units of the next layer. Different activation functions can be used, such as the logistic sigmoid function, the identity function, the hyperbolic tan function, or the rectified linear unit function (ReLU). The output layer receives the values from the last hidden layer and transforms them into output values (Pedregosa *et al.* 2011; Scikit-learn 2019). Figure 3 shows a representation of a multilayer perceptron with one hidden layer of three units in layer 2. Note that in this case a_i^l is the activation of unit i in layer l.

Training the neural network

One way to train a network is to enter the inputs, that is, the set of features $X_i = \{x_i|x_1,x_2,...,x_m\}$ in an MLP, take the output values (the hypotheses $h_\Theta(x)$) and compare them with the known results, y. A loss function can be expressed by Eq. 2 when several sets of inputs have been processed. The lower the value of the loss function, the more accurate is the network, and as can be seen in Fig. 2, the assumptions depend on the weights Θ of the units.

$$Loss = \frac{|y - h_\Theta(x)|^2}{2} = \frac{1}{2n}\sum_{n=1}^{N}\sum_{K=1}^{K}(y_k - a_k^L)^2, \tag{2}$$

where K is the number of units in the last layer, N is the number of training sets, n is the training set index, and L is the last layer. To maximize the prediction of the hypothesis, the loss function must be minimized. While the error $(y - h_\Theta)$ in the output layer is easy to check, the error in the hidden layers is not clear because the training data do not say what value the hidden nodes should have. However, it turns out that it is possible to backpropagate the error of the output to the hidden layers. The backpropagation algorithm used for learning in multilayer networks can be summarized as follows: calculate loss values $(y - h_\Theta)$ for the output units using the error observed from the output layer. Repeat this step for each layer of the network until the first hidden layer is reached. When the first hidden layer is reached, propagate the loss values back to the previous layer, and update the weights between the two layers (Russell and Norvig 2016).

Regularization

When the neural network has many features, and a high-order polynomial is used to fit the training set, the result may be overfitting. It is common to introduce restriction conditions by mean of a penalty function to the objective function, meant to include extra information. Such procedure helps to solve ill-posed problems or to prevent overfitting by ensuring the smoothness of the solution balancing complexity and accuracy (Marwala and Lagazio 2011). This procedure is named regularization, and it is penalized by the weight $\Theta_{j,i}$, multiplying a parameter α, which can be expressed by Eq. 3:

$$J(\Theta) = \frac{1}{2n}\sum_{n=1}^{N}\sum_{k=1}^{K}(y_k - a_K^L)^2 + \frac{\alpha}{2}\cdot\sum_{l=1}^{L-1}\sum_{i=1}^{S_l}\sum_{j=1}^{S_{l+1}}(\Theta_{j,i}^l)^2, \tag{3}$$

where $J(\Theta)$ is the loss function, L is the total number of layers in the network, S_L is the number of units (not counting the bias unit) in layer l, and K is the number of output units, as previously mentioned.

A high value of α tends to reduce the variance of the network objective function and imposes more biases to the network prediction. If changing regularization values does not change the accuracy of the network, then the network already has a high bias. The bias can be understood as a measure of how close to the expected value, on average, over several different training sets, an estimator is (Brown 2004). However, high bias causes underfitting in the neural network model.

Evaluating estimator performance

It is common practice when running a supervised machine learning experiment to keep some of the available data as a test set. In the procedure called cross-validation (CV), a test set should be held out for final evaluation. In the basic approach, called CV *k*-fold, the training set is divided into *k* smaller sets (folds). First, a model is trained using the folds as training data. The performance measure reported by the CV *k*-fold is then the average of the *k* calculated values (Kohavi 1995). This approach can be computationally expensive, but it does not waste a lot of data, which is a great advantage in problems where the number of samples is very small. In a second step, the resulting model is validated on the test set to compute the performance of the estimators, such as accuracy.

In this work, the neural network training set has 1093 shots registered from military exercises. The CV procedure uses 10 folds, and in each fold 70% of the data composes the training set and 30% the validation set. In the second step, additional 396 shots (27% of the total), not used in CV procedure, are used to measure the effectiveness of the neural network prediction.

METHODOLOGY: APPLICATION OF ANN TO PREDICT THE SHOOT DOWN OF THE OPPONENT

The first step in the adopted methodology correspond to the application of SODA method. As stated previously, five specialists were consulted and verified that no new information emerges after three interviews, suggesting that the five interviews have been enough to capture the relevant information. The five specialists acted together to create a unified consensual cluster. The cluster was already presented in Fig. 1.

To develop an ANN for predicting the shoot down of the opponent, the following eight influencing factors from the unified cluster were selected.

- Difference in altitude between opposing aircraft;
- Shooting distance;
- Shooting position in relation to WEZ (short, medium or long shoot);
- Minimum shooting distance;
- Closure rate;
- Shooter aircraft speed;
- Angle of the opposing aircraft in relation to the shooter aircraft, and
- The number of aircraft in the shooter's and opponent's squadron.

The database allowed selecting 1093 shoots from a military exercise database with the eight features cited. Training the MLP makes use of the backpropagation algorithm (LeCun *et al.* 2012), and three optimizers were studied to minimize the loss function. The first optimizer is the limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS) (Byrd *et al.* 1995), an optimizer in the family of quasi-Newton methods. The second one is the stochastic gradient descent (SGD) (Bottou 1991), and the third is the Adam algorithm that refers to a stochastic gradient-based optimizer, proposed by Kingma and Ba (2015). An example of an MLP for predicting the shoot down of the opponent can be seen in Fig. 4. The eight features identified in the SODA procedure feed the input layer of an MLP, and the expected answer is whether the shoot down happens.

A hyperparameter optimization for the neural network was made to select the best neural network architecture (Goodfellow *et al.* 2016). The metrics used to evaluate the results are accuracy, the proportion of correctly classified examples; precision, the proportion of true positives among instances classified as positive; recall, the proportion of true positives among all positive instances in the data; F1-score, a weighted harmonic mean of precision and recall (Dalianis 2018).

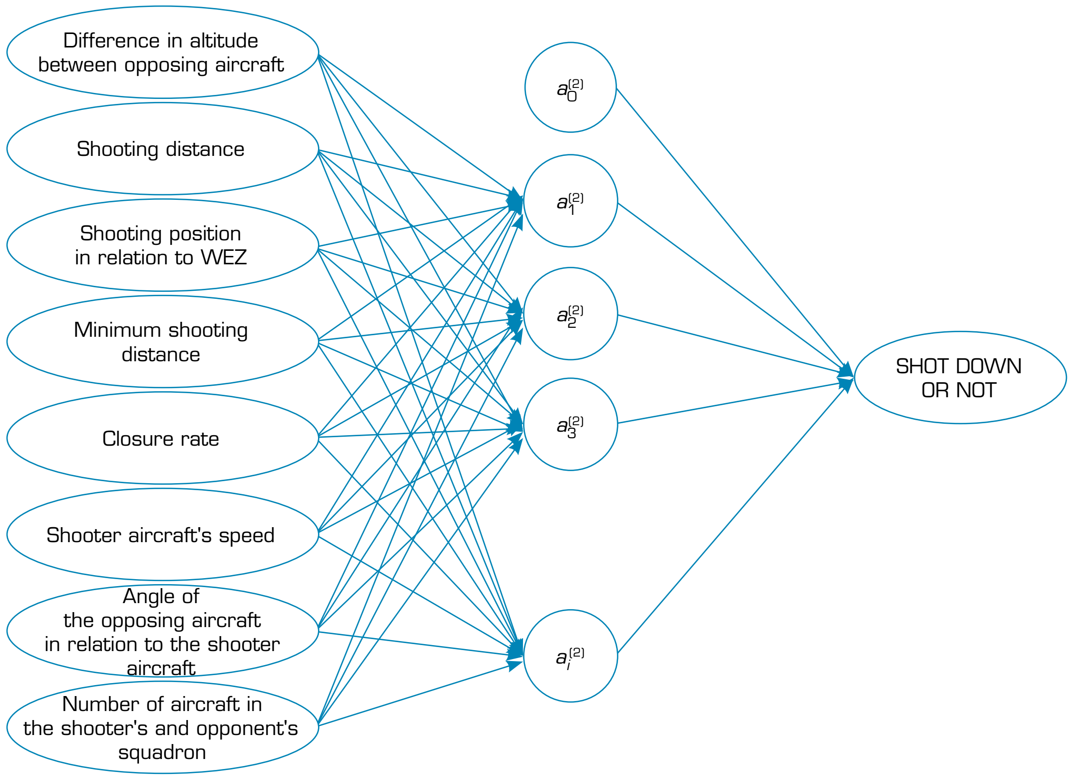

Figure 4. Example of MLP with one hidden layer for predicting the shoot down of the opponent.

The training set is balanced, that is, the number of positive and the number of negative occurrences is equivalent. The neural network aims to predict the success of a missile in each launch. The metric to be used for decision-making depends on the specific conditions of the mission and UCAV tactical situation. For instance, if the vehicle has little range and having to shoot down the enemy in the first few engagements, the total amount of true positives and true negatives can be the most important to be evaluated. On the other hand, if the vehicle has sufficient range for several engagements, its missile has a very high cost and cannot be wasted, the false-positive rate will be considered more harmful than false-negative rate. The false-positive rate is the relative frequency of no shot down when shot down is predicted, and the false-negative rate is the relative frequency of shot down when the no shot down is predicted (Riffenburgh 2012). There may also be occasions when the amount of UCAV and missiles is high, and the tactical situation determines that every shot opportunity has to be used. In this case, the amount of false negative is more harmful than false positive. The evaluation conducted in the experiments will consider the accuracy, F1-score, precision and recall metrics for the analysis.

The first phase of the optimization process started looking for a number of layers, hidden units, optimizer and activation function that optimize the cited metrics.

The ideal network architecture for a task must be found via experimentation guided by monitoring the validation set error (Goodfellow *et al.* 2016). Functions representable with a deep rectifier net can require an exponential number of hidden units with a shallow (one hidden layer) network, according to Montúfar *et al.* (2014). Therefore, in the present work, a different number of hidden units were experimented with for the first hidden layer: 8, 32, and 64. Then, additional hidden layers were added (each one with 8, 16, 32, and 64 hidden units) to evaluate their effect on the result.

The second phase of the hyperparameter optimization was to choose the best regularization. The best neural network architectures obtained in the first phase was trained with different regularizations (α = 20, 15, 10, 5, 2, 1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.01, 0.005, 0.001, 0.0001).

The third and final phase re-evaluated the number of hidden layers and units after choosing the optimizer and the activation function for the hidden layer as a function of a subset of regularization parameters.

After the hyperparameter optimization, the best architectures selected will have a performance comparison with a testing set with 396 shots previously selected and not used in the training set, as mentioned previously.

RESULTS AND DISCUSSION

The main results obtained to define the neural network architecture for predicting the shoot down of the opponent in the BVR air combat, considering the aspects discussed previously, are presented and discussed. The neural network architecture is defined in three steps: the evaluation of the number of layers and hidden units, assuming a pre-defined regularization parameter; selection of the optimizer, the activation function, and the regularization parameter; and finally, the number of hidden units is reviewed, considering the best three parameters found in the second step.

Selecting the number of layers, hidden units, optimizer and activation function

To select the best number of layers, hidden units, optimizer and activation function, the neural network was trained with different optimizers (Adam, SGD and L-BFGS), different activation functions in the hidden layer (ReLU, logistic sigmoid function, identity function and hyperbolic tangent function), $\alpha = 1$, and different number of layers and hidden units. Only the best four configurations are presented in Table 2, Table 3, and Table 4. In each table, the best result is highlighted in bold characters.

Table 2. Performance metrics from neural network architecture: Adam optimizer, ReLU, $\alpha = 1$, different number of layers and hidden units.

Hidden units	Accuracy	F_1-score	Precision	Recall
8	0.775	0.772	0.772	0.775
32	0.776	0.771	0.773	0.776
64	**0.780**	**0.775**	**0.777**	**0.780**
64, 8	0.771	0.767	0.768	0.771
64, 16	**0.780**	**0.776**	**0.777**	**0.780**
64, 32	0.775	0.771	0.772	0.775
64, 64	0.777	0.772	0.774	0.777
64, 32, 8	0.777	0.773	0.774	0.777
64, 32, 16	0.774	0.770	0.771	0.774
64, 32, 32	0.774	0.770	0.771	0.774
64, 32, 64	0.761	0.757	0.757	0.761
64, 32, 8, 8	0.772	0.769	0.769	0.772
64, 32, 8, 16	0.769	0.765	0.765	0.769
64, 32, 8, 32	0.767	0.762	0.763	0.767

Table 3. Performance metrics from neural network architecture: Adam optimizer, hyperbolic tangent function, α = 1, different types of layers and hidden units.

Hidden units	Accuracy	F_1-score	Precision	Recall
8	0.772	0.769	0.769	0.772
32	0.772	0.768	0.769	0.772
64	0.774	0.770	0.771	0.774
64, 8	0.766	0.762	0.762	0.766
64, 16	0.773	0.769	0.770	0.773
64, 32	0.774	0.770	0.771	0.774
64, 64	0.773	0.769	0.770	0.773
64, 32, 8	0.777	0.773	0.774	0.777
64, 32, 16	0.775	0.771	0.772	0.775
64, 32, 32	0.777	0.773	0.774	0.777
64, 32, 64	0.772	0.769	0.769	0.772
64, 32, 8, 8	0.780	0.776	0.776	0.780
64, 32, 8, 16	**0.777**	**0.773**	**0.774**	**0.777**
64, 32, 8, 32	0.774	0.770	0.771	0.774

Table 4. Performance metrics from neural network architecture: L-BFGS optimizer, logistic sigmoid function, α = 1, different types of layers and hidden units.

Hidden units	Accuracy	F_1-score	Precision	Recall
8	0.776	0.771	0.773	0.776
32	0.777	0.773	0.774	0.777
64	0.778	0.774	0.775	0.778
64, 8	0.771	0.767	0.768	0.771
64, 16	0.775	0.771	0.772	0.775
64, 32	0.775	0.771	0.772	0.775
64, 64	**0.780**	**0.777**	**0.777**	**0.780**
64, 32, 8	0.683	0.630	0.699	0.683
64, 32, 16	0.764	0.759	0.760	0.764
64, 32, 32	0.625	0.481	0.390	0.625
64, 32, 64	0.679	0.628	0.686	0.679
64, 32, 8, 8	0.625	0.481	0.390	0.625
64, 32, 8, 16	0.625	0.481	0.390	0.625
64, 32, 8, 32	0.625	0.481	0.390	0.625

The best neural network architectures in the four cases corresponds to: Adam optimizer, ReLU, α = 1, one layer and 64 hidden units; Adam optimizer, ReLU, α = 1, two hidden layers with 64, 16 units; Adam optimizer, hyperbolic tangent function, α = 1, four hidden layers with 64, 32, 8, 8 hidden units; L-BFGS optimizer, logistic sigmoid function, α = 1, two hidden layers with 64, 64 units. These neural network configurations are used in the next step.

Selecting the regularization

For each neural network architecture selected in the previous phase, 19 different values of the regularization parameter α were considered: α = {20, 15, 10, 5, 2, 1, 0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.01, 0.005, 0.001, 0.0001}.

Figures 5 to 8 present the performance metrics for each architecture cited in the previous subsection. Again, bold characters indicate the best result for each case. Note that the accuracy and recall response curves are superimposed on all figures.

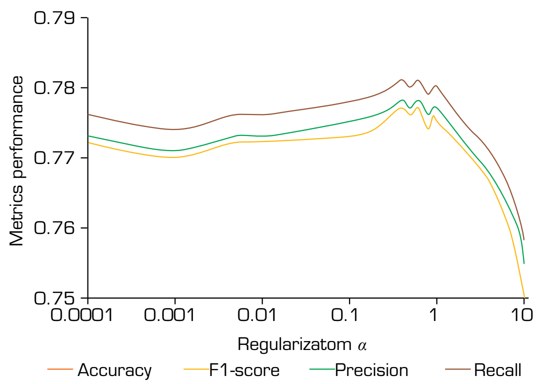

Figure 5. Performance metrics for different regularizations: Adam optimizer, ReLU, one hidden layer with 64 units.

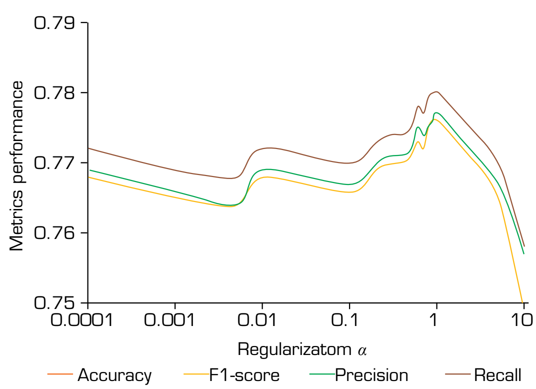

Figure 6. Performance metrics for different regularizations: Adam optimizer, ReLU, and two hidden layers with 64 and 16 units, respectively.

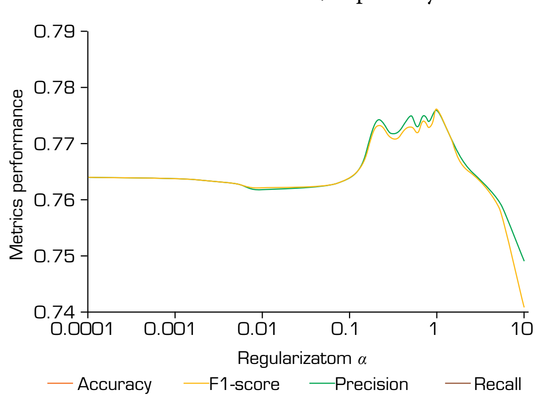

Figure 7. Performance metrics for different regularizations: Adam optimizer, hyperbolic tangent function, and four hidden layers with 64, 32, 8 and 8 units, respectively.

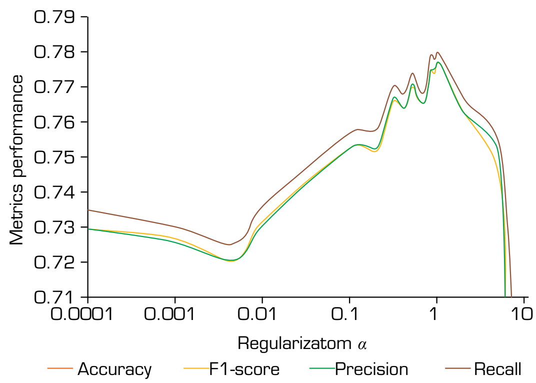

Figure 8. Performance metrics for different regularizations: L-BFGS optimizer, logistic sigmoid function, and two hidden layers with 64 and 64 units, respectively.

The tuning of the regularization parameters based on this data set shows that the four combinations of optimizer and activation function maintained the same performance metrics in predicting the best conditions for shoots, except the first architecture that showed a little improvement with α = 0.4 and α = 0.6 (Fig. 5) obtaining 78.1% accuracy, 77.7% F_1-score, 77.8% precision, 78.1% recall.

From the results obtained so far, the third and last phase of the network architecture optimization evaluates the number of hidden layers and the number of units again as a function of the α = 0.4 and α = 0.6 in the Adam optimizer, ReLU, one hidden layer with 64 units. The tests showed no improvement in the results.

A short discussion of the cross-validation results

The best five architectures found for the MLP with a backpropagation learning algorithm are:

- Neural network 1 (NN1): Adam optimizer, ReLU, one hidden layer with 64 units and α = 0.4;
- Neural network 2(NN2): Adam optimizer, ReLU, one hidden layer with 64 units and α = 0.6;
- Neural network 3 (NN3): Adam optimizer, ReLU, and two hidden layers with 64, 16 units and α = 1;
- Neural network 4 (NN4): Adam optimizer, hyperbolic tangent function, and three hidden layers with 64, 32, 8, 8 units and α = 1;
- Neural network 5 (NN5): L-BFGS optimizer, logistic sigmoid function, and two hidden layers with 64, 64 units and α = 1.

The best performance metrics obtained by using the training set is 78.1% of accuracy, 77.7% (+/– 1%) of F1-score, 77.8% (6%) of precision and 78.1% of recall.

The performance metrics of the best five architectures of the neural networks with the training set showed very similar results.

Performance comparison with a test set

An additional test using an independent data set, built with 396 shots not included in the training phase were used to reevaluate the neural networks, allowing to verify which one would better generalize new data, i.e., make better predictions. Table 5 shows the results obtained and Table 6 presents the confusion matrix of each architecture. It can be noticed that the performance of the five ANN with the test set was inferior to the one obtained in the CV procedure, as measured by the four metrics. The differences occur due to some overfitting of the training data set which are almost inevitable due to the hyperparameters tunning, making the training scores optimistic. One way to minimize this problem is to use a bigger training data set, which it is not available at this time.

Table 5. Performance metrics for different ANN with their predictions of testing set.

Regularization	Accuracy	F_1-score	Precision	Recall
NN1	0.737	0.636	0.758	0.548
NN2	0.735	0.632	0.756	0.542
NN3	0.742	0.643	0.767	0.554
NN4	0.740	0.633	0.774	0.536
NN5	0.730	0.619	0.757	0.524

Table 6. Best confusion matrix of the five ANN architectures.

Predicted by NN1					
		1.0	0.0	True positive	91
Actual	1.0	91	75	False positive	29
	0.0	29	201	False negative	75
				True negative	201
Predicted by NN2					
		1.0	0.0	True positive	90
Actual	1.0	90	76	False positive	29
	0.0	29	201	False negative	76
				True negative	201
Predicted by NN3					
		1.0	0.0	True positive	92
Actual	1.0	92	74	False positive	28
	0.0	28	202	False negative	74
				True negative	202
Predicted by NN4					
		1.0	0.0	True positive	89
Actual	1.0	89	77	False positive	26
	0.0	26	204	False negative	77
				True negative	204
Predicted by NN5					
		1.0	0.0	True positive	87
Actual	1.0	87	79	False positive	28
	0.0	28	202	False negative	79
				True negative	202

The five neural networks had very similar results in their performance, both with the validation set and the test set. The false negative rate was relatively high for all neural networks with the test set, so these neural networks should preferably be used on missions in which the total amount of true positives and true negatives must be the most important to be assessed, or on missions in which the false positive rate will be considered more damaging than the false negative rate.

The training time of the ANN, which is done just once, with the data set used was 12.5 s. After training the ANN, the measured response time is 5.3×10^{-4} s for each input set, i.e., faster than the usual human time response. These data were obtained using one core of an i7 processor. The trained ANN can also be used in an onboard system without loss of time response.

CONCLUSION

Beyond visual range air combat has a very complex decision-making process because the scenario is dynamic, changes every second, and the pilot must consider many factors to update his decision. Nowadays, many decisions have become automated to increase the probability of pilot survival. The most advanced air forces are already using UCAV instead of manned aircraft on some missions. Thereby, it is important to develop systems that make decisions automatically in a short time and with a reasonable probability of success.

This paper explores an ANN trained using a database from military exercises containing data from onboard sensor systems at the moment of launching a missile. The database is very rich, providing data from different scenarios with different aircraft numbers on both sides. The architecture of the ANN to predict the shoot down of the opponent was presented. Strategic options development and analysis method is used to evidence the relevant input data.

Considering the richness and complexity of the environments registered in the database, and the simplicity of the ANN architecture, the success probability of shooting down the opponent, based only on the onboard sensors data, is high, about 78% for all the performance metrics computed of using the training set and 74% of accuracy of using the testing set. This is a simple system, easy to program and effortless to implement in the UCAV airborne fire control computer. The trained ANN response time is low, of the order of 5×10^{-4} s, and acceptable for real-time applications. Future works could aim to refine the analysis of the factors involved in BVR combat and analyze the parameters that would increase the probability of success provided by the ANN.

AUTHORS' CONTRIBUTION

Conceptualization: Lima Filho GM and Passaro A; **Data Curation:** Lima Filho GM; **Formal Analysis:** Lima Filho GM and Passaro A; **Funding Acquisition:** Passaro A; **Investigation:** Lima Filho GM; **Methodology:** Lima Filho GM and Passaro A; **Resources:** Lima Filho GM; **Supervision:** Passaro A; **Validation:** Lima Filho GM, Passaro A and Medeiros FLL; **Visualization:** Lima Filho GM and Passaro A; **Writing – Original Draft Preparation:** Lima Filho GM; **Writing – Review & Editing:** Passaro A and Medeiros FLL.

DATA AVAILABILITY STATEMENT

Not applicable.

FUNDING

Conselho Nacional de Desenvolvimento Científico e Tecnológico
[https://doi.org/10.13039/501100003593]
Grant No. 311324/2016-9 and 307691/2020-9

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
[https://doi.org/10.13039/501100002322]
Finance Code 001

ACKNOWLEDGEMENTS

The authors would like to thank Comando de Preparo of the Brazilian Air Force for the support for this research. Also, the authors would like to thank Dr. Olympio Lucchini Coutinho coordinator of the Postgraduate Program in Operational Applications from the Aeronautical Technology Institute for the collaboration.

REFERENCES

Ackermann F, Eden C (2010) Strategic options development and analysis. In: Reynolds M, Holwell S, editors. Systems Approaches to Managing Change: A Practical Guide. London: Springer. p. 135-190. https://doi.org/10.1007/978-1-84882-809-4_4

Alpaydin E (2014) Introduction to machine learning. Cambridge: MIT.

Bottou L (1991) Stochastic gradient learning in neural networks. Proceedings of the Neuro-Nîmes 91(8):12.

Brain G (1999) Aerial Combat Manoeuvre Identification - A Feasibility Study of an Intelligent Decision Support System. Loughborough University. Unpublished.

Brown G (2004) Diversity in Neural Network Ensembles (Doctoral thesis). Birmingham: University of Birmingham.

Byrd RH, Lu P, Nocedal J, Zhu C (1995) A Limited Memory Algorithm for Bound Constrained Optimization. SIAM J Sci Comput 16(5):1190-1208. https://doi.org/10.1137/0916069

Dalianis H (2018) Clinical Text Mining. Cham: Springer. Evaluation Metrics and Evaluation; p. 45-53. https://doi.org/10.1007/978-3-319-78503-5_6

Eden C, Ackermann F (2001) SODA: The principles. In: Rosenhead J, Mingers J, editors. Rational analysis for a problematic world revisited: Problem structuring methods for complexity, uncertainty and conflict. Hoboken: Wiley. p. 21-41.

Galante A (2010) Promessa e realidade: o combate ar-ar BVR. Poder Aéreo. [accessed Apr 3 2020]. https://www.aereo.jor.br/2010/08/22/promessa-e-realidade-o-combate-ar-ar-bvr/

Georgiou I (2011) Cognitive Mapping and Strategic Options Development and Analysis (SODA). In: Cochran JJ, Cox Junior LA, Keskinocak P, Kharoufeh JP, Smith JC, editors. Wiley Encyclopedia of Operations Research and Management Science. Hoboken: Wiley. p. 679-688. https://doi.org/10.1002/9780470400531.eorms0974

Gilberto HB (2017) Combat beyond visual range: a complex environment for decision-making. Revista da UNIFA 30(1):15-22.

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. Cambridge: MIT.

Kingma DP, Ba J (2015) Adam: A Method for Stochastic Optimization. Paper presented 3rd International Conference for Learning Representations. ICLR; San Diego, California, United States.

Kohavi R (1995) A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence; 1995 Aug 20; Montreal, Quebec, Canada. San Francisco: Morgan Kaufmann; 1995.

LeCun YA, Bottou L, Orr G.B, Müller K-R (2012) Efficient BackProp. In: Montavon G, Orr GB, Müller K-R, editors. Neural Networks: Tricks of the Trade. Berlin: Springer. p. 9-48. https://doi.org/10.1007/978-3-642-35289-8_3

Lei X, Huang A, Zhao T, Su Y, Ren C (2018) A New Machine Learning Framework for Air Combat Intelligent Virtual Opponent. J Phys: Conf Ser 1069:012031. https://doi.org/10.1088/1742-6596/1069/1/012031

Liang X, Huang J, Zhou Y (2010) Study on the effectiveness evaluation method of formation Beyond Visual Range air combat based on genetic BP Neural Network. Proceedings of the 2010 Asia-Pacific International Symposium on Aerospace Technology, APISAT 2010; 2010 Sep 1; Xi'an, China. Xi'an: Northwestern Polytechnical University; 2015.

Ma Y, Wang G, Hu X, Luo H, Lei X (2019) Cooperative Occupancy Decision Making of Multi-UAV in Beyond-Visual-Range

Air Combat: A Game Theory Approach. IEEE Access 8:11624-11634. https://doi.org/10.1109/ACCESS.2019.2933022

Macedo LH (2017) Alocação coordenada de múltiplos alvos aéreos utilizando heurística especializada (Master thesis). São José dos Campos: Instituto Tecnológico de Aeronáutica. In Portuguese.

Marttunen M, Lienert J, Belton V (2017) Structuring problems for Multi-Criteria Decision Analysis in practice: A literature review of method combinations. Eur J Oper Res 263(1):1-17. https://doi.org/10.1016/j.ejor.2017.04.041

Marwala T, Lagazio M (2011) Militarized Conflict Modeling Using Computational Intelligence. Cham: Springer. https://doi.org/10.1007/978-0-85729-790-7

Meng D, Wang Y, Chen Y, Zhong L (2014) Tactics Decision-Making Based on Granular Computing in Cooperative Team Air Combat. In: Wen Z, Li T, editors. Practical Applications of Intelligent Systems. Heidelberg: Springer. p. 915-925. https://doi.org/10.1007/978-3-642-54927-4_88

Mingoti SA, Lima JO (2006) Comparing SOM neural network with Fuzzy *c*-means, *K*-means and traditional hierarchical clustering algorithms. Eur J Oper Res 174(3):1742-1759. https://doi.org/10.1016/j.ejor.2005.03.039

Montúfar G, Pascanu R, Cho K, Bengio Y (2014) On the Number of Linear Regions of Deep Neural Networks. arXiv preprint:1402.1869.

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, *et al.* (2011) Scikit-learn: Machine Learning in Python. J Mach Learn Res 12:2825-2830.

Riffenburgh RH (2012) Statistics in Medicine. Cambridge: Academic Press. Chapter 10, Risks, Odds, and ROC Curves; p. 203-219. https://doi.org/10.1016/B978-0-12-384864-2.00010-X

Rodin EY, Amin SM (1992) Maneuver prediction in air combat via artificial neural networks. Comput Math with Appl 24(3):95-112. https://doi.org/10.1016/0898-1221(92)90217-6

Rodin EY, Lirov Y, Mittinik S, McElhaney BG, Wilbur L (1987) Artificial Intelligence in Air Combat Games. In: Yavin Y, Pachter M, Rodin EY, editors. Pursuit-Evasion Differential Games. Oxford: Pergamon. p. 261-274. https://doi.org/10.1016/B978-0-08-034862-9.50026-8

Russell SJ, Norvig P (2016) Artificial Intelligence: A Modern Approach. London: Pearson.

Scikit-learn (2019) User Guide. [accessed Jun 7 2019]. https://scikit-learn.org/stable/modules/neural_networks_supervised.html

Strauss AL, Corbin JM (1998) Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory. Thousand Oaks: Sage Publications.

Suresh S, Sundararajan N, Savitha R (2013) Supervised Learning with Complex-valued Neural Networks. Cham: Springer. https://doi.org/10.1007/978-3-642-29491-4

Suseno PAP, Sasongko RA (2016) Development of Air Combat Effectiveness Simulation and Analysis Scheme for beyond Visual Range (BVR) Case. Appl Mech Mater 842:329-336. https://doi.org/10.4028/www.scientific.net/AMM.842.329

Sutton RS, Barto AG (2018) Reinforcement Learning: An Introduction. Cambridge: MIT.

Toubman A, Roessingh JJ, Spronck P, Plaat A, van den Herik J (2016) Rapid Adaptation of Air Combat Behaviour. Front Artif Intell Appl 285:1791-1796. https://doi.org/10.3233/978-1-61499-672-9-1791

Wanyang W, Yiqin Y, Jinxing L, Yong S (2020) Organize into Teams a Target Allotment a Problem according to the Threat

Situation and the Weapon Effect. Paper presented 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC). IEEE; Dalian, China. https://doi.org/10.1109/IPEC49694.2020.9115193

Wu LY, Sun YQ, Ji JY (2013) Study on Threat Assessment Based on Rough Ret and Support Vector Machines for Regression in BVR Air Combat. Appl Mech Mater 419:228-234. https://doi.org/10.4028/www.scientific.net/AMM.419.228

Ximeng X, Rennong Y, Yang Y (2019) Threat Assessment in Air Combat Based on ELM Neural Network. Paper presented 2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA). IEEE; Dalian, China. https://doi.org/10.1109/ICAICA.2019.8873461

Yu H, Wu TA (2011) A Model of Tactic Threat Assessment in PAAIS. Appl Mech Mater 130-134:204-207. https://doi.org/10.4028/www.scientific.net/AMM.130-134.204