Spectrum - The Journal of Operational Applications in Defense Areas

Journal homepage: https://spectrum.ita.br

Operational Analysis and Logistics Engineering: Simulation and decision support

Comparative Performance Analysis of Two Multi-Aerial Threat Evaluation Algorithms

Humberto Baldessarini Pires[1], Lamartine Nogueira Frutuoso Guimarães[1,2]

[1]Instituto Tecnológico de Aeronáutica, São José dos Campos/SP – Brazil

[2]Instituto de Estudos Avançados, São José dos Campos/SP – Brazil

Article Info

Article History:

Received	07 May 2025
Revised	28 June 2025
Accepted	23 July 2025
Available online	23 September 2025

Keywords:

Threat evaluation
Air defense operations
Real-time
Machine learning
Artificial neural network

E-mail adresses:

humbertohbp@ita.br
lamar.guima@gmail.com

Abstract

In modern aerial defense operation, the evaluation of potential threats is of paramount importance for effective response strategies, particularly when such assessment is performed in real-time. This study presents a comparative analysis of an algorithm developed by the authors, and referred to as DM, and a Markov chain-based approach (MC) in terms of prediction accuracy, execution time, and processing capacity. Notably, DM consistently achieved higher accuracy until simulation time 1350, despite both methods utilizing the same Artificial Neural Network architecture. Additionally, DM exhibited superior execution time and processing capacity, handling a maximum of 89 threats within a one-second timeframe, while MC processed 10 threats. Based on this, it can be asserted that DM meets the requirements for real-time threat evaluation. The results can be attributed to DM's simplified methodology, enabling more accurate and distinct predictions.

I. INTRODUCTION

The rapid advancements in weapons and information technology have significantly complicated aerial defense operations [1]. The task of safeguarding Defended Assets (DAs) from hostile air-breathing threats (ABTs) such as fighter planes has evolved into a complex endeavor due to the advancement of these threats [2]. Given that these ABTs might neutralize or annihilate critical structures and areas, it is vital to scrutinize the tactical situation in real time [3]. Real-time evaluation affords the required flexibility and agility to navigate these dynamic challenges, enabling air defense systems to swiftly adapt their defense strategies and effectively neutralize emerging threats [1].

The challenge of effectively using battlefield target detection information to assist commanders in decision-making has gained prominence in research [4]. Friendly forces utilize radars for gathering information and identifying incoming ABTs. Using this radar information, counter-air operations are executed through Threat Evaluation (TE) and weapons allocation [5]. A comprehensive evaluation of enemy threats is pivotal for increasing survivability and gaining battlefield advantage [4].

The TE process in anti-aircraft defense systems quantifies threat levels based on potential combat abilities and intentions of enemy targets [4]. In the end, the objective of TE is to prioritize engagements for optimal Weapon Target Allocation (WTA) by assigning Threat Values (TVs) to detected ABTs [5]. This procedure allows the defending force to allocate weaponry correctly against hostile targets, considering the limited availability of weaponry in real-world scenarios [6].

Predicting a target's state and understanding its intentions in advance enhances the effectiveness of defense decision-making systems, augmenting operational efficiency, conserving defense resources and playing a critical role in future command and control systems [7]. Thus, precise TE is essential in enhancing the accuracy of the defensive decision-making process to effectively counter the most significant threats [8].

Among the techniques used to model the TE process and quantify the degree of threat posed by ABTs, Machine Learning-based methods hold significant potential. They can accommodate the dynamic nature of aerial defense scenarios, a critical aspect in predicting time-sensitive information that other methods may overlook [9].

Ref. [5] partially achieved this objective. Assessments were conducted using information gathered by radar in a scenario where ABTs were detected. The ABTs' trajectories were modeled using a Markov chain-based algorithm that

implemented two distinct Artificial Neural Networks (NNs) [10]. The executed simulations reached a prediction accuracy of 35%. Therefore, there is a need for more effective TE methods that balance accuracy with prompt execution of the related tasks.

In this regard, the present study proposes an improved version of the approach delineated in [5] aiming to optimize the identification of potential targets of multiple ABTs by predicting their attack routes under real-time constraints. The model employs two distinct Machine Learning (ML) techniques to handle the ABTs' attributes. The initial technique (function α) predicts the subsequent position of the aircraft, while the second one (function β) outputs predicted attributes of the ABT in the succeeding phase. This procedure is iteratively carried out until the TAs of all ABTs are estimated for each given time interval.

II. RELEVANT LITERATURE

In the past, there was limited literature on the topic of threat assessment compared to extensive research on WTA [3]. However, in recent years, there has been a significant increase in publications focused on TE. These studies can be classified according to the employed methods, although there is a dearth of explicit criteria and inconsistency in categorization among authors. This selective citation presents relevant research on threat assessment, discussing the benefits and limitations of different methods without attempting to classify them into specific categories.

Two traditional techniques predict the arrival times of individual ABTs to a Target Asset (TA) by considering attribute information. These methods assign a high TV when the predicted arrival time is imminent. For example, [11] proposed the Closest Point of Approach (CPA) concept, while [12] utilized the Radial Speed Vector (RSV) method. However, both CPA and RSV methods have a critical constraint as they assume that the objective of an ABT is the TA with the shortest time to arrival.

Other techniques are commonly used to model the TE process and quantify the degree of danger posed by ABTs, such as Lanchester's equation [13], multi-attribute decision making [14], game theory [15], and the threat index method [4]. The rule-based approach has also been explored in [16], where experienced air defense officers select variables and develop TE rules. Fuzzy approaches [17] and Bayesian networks [18] have been proposed to combine expert knowledge with inference rules. While these methods can ensure accuracy, real-time performance becomes a challenge. Another significant limitation of these approaches is their dependency on the knowledge of subject matter experts, adding a subjective element to the process [5].

Conversely, Artificial Intelligence-based techniques have the potential to address the challenge of real-time performance and provide accurate predictions in dynamic aerial defense environments [19], without relying on expert experience [5]. For instance, supervised ML techniques provide efficient assessment processes with reduced execution times by exploring connections between features and labels [19]. Examples include NN [7] and Support Vector Machine (SVM) [20]. NN offers self-learning and reasoning abilities [21], while SVM is suitable for small sample prediction [9].

As previously discussed, [5] introduced a methodology for performing TE on ABTs using radar data within a context where hostile ABTs are targeting multiple DAs. This method is comprised of TA prediction and TE. For the TA prediction, they proposed an algorithm where an ABT's movement trajectory was modeled using a Markov chain, while the one-step transition probability of the ABT was characterized by NNs. The TV for each ABT was evaluated by factoring in the distance and weight of the DA identified by the TA prediction algorithm. They ran an air-attack simulation that incorporated 406 ABTs originating from fifteen different locations and targeting twenty friendly DAs.

Despite the setup facilitating prediction of final ABT TAs by calculating probabilities of all potential attack routes without expert reliance, the 35% accuracy achieved in simulations might undermine the subsequent WTA task, especially with typically limited weaponry. Also, the study didn't assess the algorithm's execution time, a crucial metric in evaluating anti-aircraft defense threat assessment algorithms. Quick, efficient responses are pivotal in scenarios requiring real-time decisions, handling large data volumes, optimizing limited resources use, and adapting to evolving threats dynamically. Therefore, time-efficient algorithms are fundamental to anti-aircraft defense efficacy.

III. THE PROPOSED MTHOD

This research executes an algorithm capable of processing radar-provided information about ABTs (known as targets) at specific time intervals. Utilizing two ML techniques, the algorithm aims to predict the TV of each ABT by performing this process for every ABT and time interval from initial detection until the end of the attack route. Fighter aircraft are used as target types, and parameters such as position, speed, and heading (indicative of the target's intent), are obtained from a radar located in friendly territory.

The battlefield was compartmentalized into hexagonal cells for TA prediction, using the concept of cell time defined in [5], where the state is determined by the cell the ABT currently occupies. As an aircraft moves from one cell to another, the cell time, represented by η, increases by 1. However, this state transition time η is different from the radar scanning time t, meaning that for the same cell time η, there could be multiple scanning times t. This division of space into hexagonal cells allows for the tracking of changes in ABT attributes over its route for a given cell, aiding ML techniques in predicting the ABT's next cell and its attributes.

The conducted experiments were based on the following assumptions: (1) ABTs adhere to pre-determined operationally feasible routes; (2) ABTs are continuously detected with no constraints on radar detection performance; and (3) the combat radius R_i of an ABT is determined by the maximum number of cells it can traverse through its route. The importance of considering the relative position of an aircraft within its current cell is illustrated in Fig. 1. This relative position, indicated by points A, B, and C within cell c12, can influence the aircraft's next move, even when the direction or heading remains the same (150°).

The relative position also dictates the attribute values of

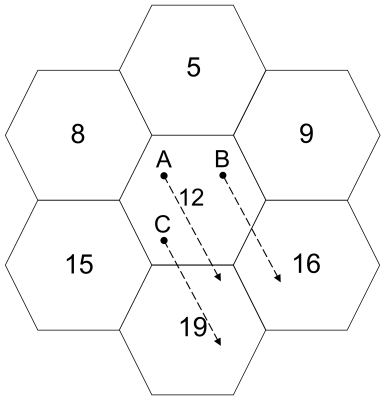

Fig. 1. Effect of relative location on the subsequent state and moving direction [5].

the ABT in the next cell, enabling ML techniques to gain deeper insights into the correlations among these parameters and ultimately enhancing the accuracy of subsequent cell time predictions.

The directions of movement are established based on the angular position between the center of the current cell and the center of each adjacent cell. Possible movements occur at angles of 0°, 60°, 120°, 180°, 240°, and 360°. The likelihood of an aircraft moving to an adjacent cell is expressed as a function of the angle between the centers of these cells, represented by the probability vector $\mathbf{P} = [p_0, p_{60}, p_{180}, p_{240}, p_{300}]$. Each element of vector $\mathbf{P}$ is assigned a class, with p_0 as class 0 and p_{300} as class 5.

The algorithm uses radar information to create an attribute vector $[x_i^t, y_i^t, v_i^t, \theta_i^t]$ for an ABT i at radar scanning time t, excluding altitude (z) information. The attribute vector $\chi_{i,\eta}$ is defined by (1) and includes the ABT's average attributes as it passes through the current cell and its relative position to the center of the current cell.

$$\chi_{i,\eta} = [\tilde{x}_{i,\eta}, \tilde{y}_{i,\eta}, \bar{v}_{i,\eta}, \bar{\theta}_{i,\eta}], \tag{1}$$

where $\tilde{x}_{i,\eta}$ and $\tilde{y}_{i,\eta}$ denote the aircraft's relative position in reference to the center of the current cell (x_c, y_c). Equations (2) and (3) depict the mathematical expressions used to calculate the aircraft's relative position at each radar scanning instance.

$$\tilde{x} = \bar{x} - x_c \tag{2}$$

$$\tilde{y} = \bar{y} - y_c \tag{3}$$

For the computation of the ABT's average speed and heading within a single cell, the current cell of the ABT must first be identified. The attribute vector $\bar{\chi}_{i,\eta+1}$ at the cell time $\eta + 1$ is then predicted using the function β as defined in (4).

$$\chi_{i,\eta+1} = \beta(\chi_{i,\eta}) \tag{4}$$

The necessary data to predict the subsequent cell that the ABT will occupy depends on the attribute vector $\chi_{i,\eta}$ of the ABT and its threat factor. The ABT's consistent movement is towards a DA for initiating an attack, and the rate of proximity to the DA is acknowledged as the threat factor $\delta_{i,m}$ on DA m, expressed by (5).

$$T_{i,m,\eta} = \frac{\bar{v}_i \cos(h_{i,m,\eta})}{d_{i,m,\eta}}, \tag{5}$$

where $h_{i,m,\eta}$ denotes the relative angle between the course of the ABT and DA, while $d_{i,m,\eta}$ signifies the distance from the ABT i to the DA m.

After the threat factors have been calculated, a novel attribute vector is formed, denoted as $\tilde{\chi}_{i,\eta} = [\tilde{x}_{i,\eta}, \tilde{y}_{i,\eta}, \bar{v}_{i,\eta}, T_{i,1,\eta}, ..., T_{i,m,\eta}]$. This updated vector incorporates the attributes from vector $\chi_{i,\eta}$ and the threat factor for an ABT related to each existing DA. The objective is to use this enriched vector in function α to predict the class of the subsequent cell the ABT is likely to occupy, as detailed in (6).

$$Class(0to5) = \alpha(\tilde{\chi}_{i,\eta}) \tag{6}$$

During the same cell time, the algorithm conducts a series of predictions until the cell time reaches the ABT's combat radius. The vector that compiles the probabilities of the predicted cells for a particular cell time is denoted by (7).

$$P_{i,\eta'} = [\pi_{i,\eta',1}, ..., \pi_{i,\eta',n}], \tag{7}$$

where $P(c_{i,\eta'} = c)$ denotes the likelihood of the ABT i being situated in a cell c at time η'. The vector $\mathbf{\Omega}$ compiles the corresponding probabilities of all potential cells that may be traversed along the ABTs' routes, with the aim of determining the target cell at the conclusion of this process. This target cell will be the one with the maximum accumulated probability. The vector $\mathbf{\Omega}$ is defined by (8).

$$\Omega = \sum_{i=\eta}^{R} P_{i,\eta'} \tag{8}$$

The algorithm computes the TV of an ABT factoring in two considerations: the weight attached to the projected DA target, gauged by the degree of potential loss or harm that friendly units would endure if the DA is eliminated or neutralized; and the proximity between the threat object and the DA, with the calculated $TV_{i,\eta}$ being inversely proportional to their distance (the $TV_{i,\eta}$ augments as the threat object approaches). These computations are encapsulated by (9).

$$TV_{i,\eta} = \frac{\omega_m}{d_{i,m,\eta}}, \tag{9}$$

where W_m is the significance given to DA m, and $d_{i,m}$ is the distance between the ABT i and the DA m, which is predicted as the target at cell time η by the algorithm.

The process conducted by the algorithm initiates as soon as ABT information is acquired from the radar and culminates with the generation of the engagement priority list. This list is derived from the TVs which are contingent upon the importance ascribed to the projected DA targets for each ABT. The threat assessment process at a given time moment t_k is outlined by Algorithm 1.

The algorithm initializes in Step 1 once it receives information from the radar. The initial engagement priority list is

Algorithm 1: Developed Algorithm

Step 1. Receive information from Radar
Generate an empty engagement priority list
Step 2. for ABT i during a time t **do**
  Create the attribute vector $[x_i^t, y_i^t, v_i^t, \theta_i^t]$
  Identify the *current_cell*
  Obtain relatives longitude and latitude, mean speed and direction
  Create $X_{i,\eta}$
  Let $\eta' \leftarrow \eta$, and $X_{i,\eta'} \leftarrow X_{i,\eta}$
  Step 3. if $\eta' < R$, then (otherwise, go to step 6) **then**
    Define $P_{i,\eta'}$
    Create Ω
    Step 4. while $\eta' < R$ **do**
      Create $P_{i,\eta'+1}$ as a zero vector
      Calculate $T_{i,m,\eta'}$
      Create $\tilde{X}_{i,\eta'+1}$
      Class (0 to 5) = $g(\tilde{X}_{i,\eta'})$
      Identify the next cell $c(d)$ as the adjacent cell c located at d
      target_cell $\leftarrow c(d)$
      $\Pi_{\eta'+1,target\_cell} \leftarrow 1$
      Perform the updates:
      $P_{i,\eta'+1} \leftarrow P_{i,\eta'} \times \frac{1}{(N_s - \eta')}$
      $X_{i,\eta'+1} \leftarrow f(X_{i,\eta'})$
      current_cell $\leftarrow$ *target_cell*
      $P_{i,\eta'} \leftarrow P_{i,\eta'} + P_{i,\eta'+1}$
      $\Omega \leftarrow \Omega + P_{i,\eta'}$
      $\eta' \leftarrow \eta' + 1$
    end while
    Step 5. Create a final probability list containing tuples (cell, probability) for each cell in which the DAs are located
    Determine the cell with the highest probability in the final probability list
    Determine the *target_DA*
    if there are multiple DAs in the *current_cell* with the highest probability in the final probability list **do**
      target_DA $\leftarrow$ DA with the highest threat factor
    end if
  end if
  Step 6. *target_DA* $\leftarrow$ DA in the *current_cell*
  if there are multiple DAs in the *current_cell* **do**
    target_DA $\leftarrow$ DA with the highest threat factor
  end if
  Step 7. Calculate *TV*
  Add the tuple (ABT, *TV*) in the engagement priority list
end for
Step 8. Normalize the engagement priority list containing the TVs of each detected ABT and sort the values in descending order
Display the engagement priority list

empty, and it will store calculated TVs for detected ABTs at scanning time t.

In Step 2, a vector including speed, direction, and position (longitude and latitude) of each ABT is created. This allows the identification of the current cells occupied by the ABTs. The current cell information is key to calculating parameters within attribute vectors used as inputs for functions α and β.

Step 3 generates two probability vectors: $P_{i,\eta'}$ indicates the probability of an ABT occupying a cell at time η', and the final probability vector Ω presents the accumulated probability for each cell in the scenario.

If the combat radius R_i of an ABT i hasn't been reached, Step 4 computes the threat factor of the ABT with respect to each DA based on the current position of the ABT using (5). Through (6) and (4), functions α and β predict the next cell and calculate the parameters of the ABT in the predicted new cell, respectively. As a result, the cell time η' increases by one, and the predicted cell becomes the ABT's current cell. This entire process is then repeated. The probability vector $P_{i,\eta'}$ accumulates the probabilities of the ABT occupying the adjacent cell in future moments as iterations (representing transitions from one cell to another) are performed. This whole step is repeated until the while condition is fulfilled.

Step 5 produces the final probability list containing tuples (cell, probability) for each cell housing DAs. The target DA is identified as the one located in the ABT's current cell, based on the highest observed probability value. In cases where multiple DAs are in the current cell, the target DA is chosen as the one with the highest threat factor.

When the combat radius R_i of an ABT i is reached, Steps 3, 4, and 5 are bypassed. The algorithm moves to Step 6, where the target DA for each ABT is selected as the one in the current cell of the ABT. In cases of multiple DAs in the current cell, the process described in the previous step is performed.

In Step 7, the ABT's TV is calculated according to (9), and a tuple with its identification and the calculated value is added to the engagement priority list generated in Step 1.

Lastly, Step 8 sorts the engagement priority list in descending order based on normalized TVs, providing the ABTs in the optimal sequence for engagement. The list is then displayed, and the algorithm stops.

IV. SIMULATION EXPERIMENTS

A. *Configurated scenario*

This study was based on a battlefield scenario measuring 370 km in width and 278 km in length. In this setting, fifteen origin points for attacking aircraft, twenty targets, and 28 waypoints were defined. This setup allowed the attacking aircraft to swiftly change directions, intending to avoid anti-aircraft defenses. The combination of these points resulted in 41 viable routes, designed to simulate a wide range of possible attack patterns. The airspace was divided into hexagonal cells, each measuring 40 km on each side.

For each simulation, 200 ABTs were generated. Waves of two, four, six, or eight new ABTs were detected on routes among the 41 routes at each time interval t. Random seeds were used to determine the number of ABTs in each wave and to select the routes for each aircraft. This method ensured the cases were distinct and allowed for a diverse evaluation of the algorithms' performances. Additionally, Gaussian noise was added to the ABTs' speed and heading to replicate realistic scenarios, leading to different flight trajectories for each ABT even on the same attack route.

From the origin point, the first-time detected ABT by the radar was assigned an initial speed of 230 m/s, which increased by 20 m/s every 300 radar scans. The final detected speeds varied due to the different lengths of routes. Fig. 2 shows the scenario including the origin points, waypoints, and the locations of the twenty TA or DAs. The importance

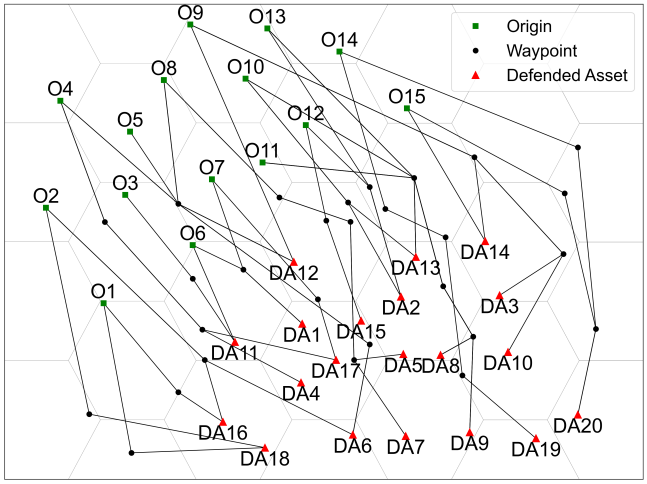

Fig. 2. The configurated scenario.

values assigned to each of the configured DAs in the utilized scenario were the same as those used in [5].

B. Evaluation of ML techniques

The performance of the developed method, referred to as DM in this study, was evaluated by comparing it to a Markov chain-based algorithm (referred to as MC). Both configurations utilized two Multi-Layer Perceptron (MLP) neural networks (Rosenblatt, 1958) in functions α and β, with specific hyperparameters detailed in Table I. The datasets were divided into a 70% training set and a 30% validation set. The evaluation metric used for function α was Accuracy (ACC), while Mean Squared Error (MSE) was employed as the evaluation metric for function β.

Table I. Hyperparameters of the set of NN employed in MC and DM configurations.

Hyperparameter	Function α (NN)	Function β (NN)
Solver	Adam	Adam
Hidden Layers	2	2
Hidden Nodes per Layer	50	50
Learning Rate	0.0003	0.0003
Activation Function	Tanh [22]	Tanh [22]

During training and validation, 200 ABTs were simulated on twenty randomly selected attack routes. This meant that roughly half of the total possible routes were used for training. The dataset for function α comprised ABT information recorded every second, the threat factors of the ABT for each DA, and the indicator of the next cell to be traversed. For function β, the dataset was a vector of ABT's relative position, mean speed, and mean heading. The inputs and outputs for the functions α and β are delineated by (6) and (4), respectively.

Table II illustrates the performance of NNs techniques in MC and DM configurations. The obtained values in the evaluation metrics of the validation sets provided confirmation that the NNs were trained without experiencing overfitting.

Table II. Performance of the NN techniques employed in MC and DM configurations.

Technique	Function α (ACC) Training / Validation Set	Function β (MSE) Training / Validation Set
NN	0.985 / 0.985	-
NN	-	7.483 / 7.292

V. RESULTS AND COMPARATIVE ANALYSIS

The research focused on assessing the efficiency of algorithms using performance metrics such as accuracy, execution time - measured from the moment ABT information was received until the creation of the engagement priority list - and processing capacity. In order to facilitate a fair comparison, both algorithms shared the same structure and functions in the common calculations between them, with the only difference being in the specifically defined approach for the predictions in the function α and the calculation of cumulative probability vectors.

The simulations were performed on a computer equipped with an Intel Core i7 10750H processor, six cores @ 2.60 GHz, and 16 Gb RAM. This computational setup provided sufficient processing power and memory capacity to ensure reliable performance evaluations of the algorithms under investigation. Twenty simulated cases were performed, each generating 200 ABTs over multiple time steps.

The accuracies generated by the two methods (MC and DM) were assessed using statistical tests. The Shapiro-Wilk [23] normality test indicated that the results did not follow a normal distribution, although the Levene test [24] assessed the equality of variances. To address that limitation, a pairwise Kruskal-Wallis H-test [25] was performed, which yielded statistically significant results.

Fig. 3 presents the mean accuracy of ABTs in flight at each simulation time for the twenty simulated cases. As the ABTs advanced along their routes, the accuracies of both methods approached convergence. It can be noted that DM outperformed MC consistently until approximately simulation time 1350, despite using the same NN architecture.

One possible reason is that DM simplifies the prediction task by focusing on a single class, while MC requires the network to assign probabilities to multiple classes. This single-class nature of DM makes it easier for the network to learn and achieve accurate predictions.

Another contributing factor could be the decision threshold employed in each predictive approach. The DM prediction function typically uses a predefined threshold to classify an instance into a specific class. This threshold can be set to optimize accuracy for a particular class, making it more effective for predicting that specific class accurately. In contrast, the MC prediction function provides probabilities for all classes, and the decision threshold might not be optimized for the specific class of interest, leading to diminished accuracy. Additionally, the single-class prediction could benefit from a less complex decision boundary compared to the multi-class prediction. By focusing on a single class, the network might be capable of learning a more distinctive and easier-to-identify decision boundary, resulting in superior predictions.

The execution times of the algorithms were evaluated in

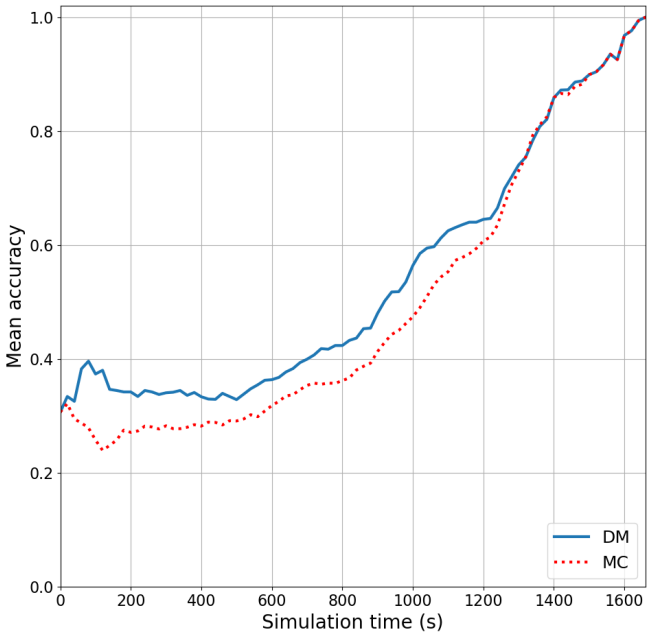

Fig. 3. Mean accuracies of the algorithms by simulation time (20 cases).

addition to the accuracies. A randomly selected case (from the cases used in the previous metric) was executed twenty times. Statistical tests confirmed significant differences in the execution times between MC and DM, shown in Fig. 4. Once again, DM demonstrated superior performance compared to MC, with a noticeable increase in execution time directly correlated with the number of ABTs in flight (indicated by the dash-dot gray line), especially for MC. This difference can be attributed to the fact that DM predicts a single class for each time step, while MC predicts probabilities for multiple adjacent cells.

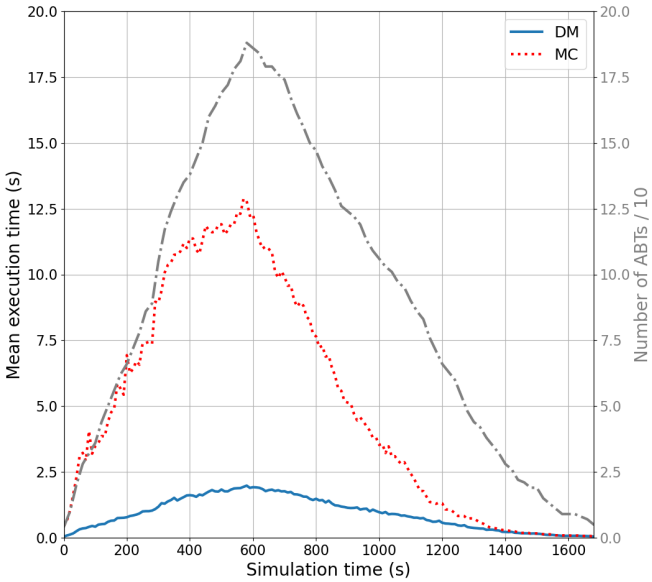

Fig. 4. Mean execution times of the algorithms by simulation time and number of ABTs/10 (selected case).

In the context of a real-world application, it is crucial that the algorithm's responses enable timely processing of the generated information by decision-makers. This ensures the effective execution of the succeeding steps in the air defense process. The evaluation of the algorithms in terms of processing capacity was based on a one-second radar scanning time interval, which served as a benchmark for comparing the performance of the algorithms in processing a certain number of ABTs.

Table III presents the average number, along with their standard deviation, of ABTs processed within the specified time interval for each simulated case. The results indicate that MC managed to process 10 ABTs within the one-second time-frame in the randomly selected case, while DM processed a maximum of 89 ABTs, showcasing its better performance. As previously explained, this can be attributed to the multi-prediction approach utilized by the MC algorithm when compared to the single-class prediction nature of DM.

Table III. Maximum number of ABTs processed within a one-second time-frame and respective standard deviations (SD).

	Number of ABTs (mean $\pm$ SD)	
Case	**MC**	**DM**
Selected case	10.0 ± 0.0	89.4 ± 1.8

VI. CONCLUSIONS

The assessment of potential threats is vital in air defense as it provides valuable insights into the ever-changing threat landscape and enables prompt responses. This study aimed to compare the performance of a developed algorithm (DM), and a Markov chain-based approach (MC). The algorithms were evaluated in terms of prediction accuracy, execution time, and processing capacity under rigorous statistical tests. To enable a fair comparison, both evaluated algorithms had similar structures and functions for common calculations. As the simulations progressed, both methods showed converging accuracies. However, DM outperformed MC until simulation time 1350, despite using the same NN architecture.

In terms of execution time, DM exhibited significantly better performance. The execution time increased as the number of ABTs processed increased, particularly for MC.

The evaluation of processing capacity was based on a one-second radar scanning time interval. DM surpassed MC in terms of ABT processing capacity, with DM processing a maximum of 89 ABTs compared to MC's processing of 10 ABTs within the one-second time-frame in the selected case.

The superior performance of the developed algorithm across all evaluated metrics can be attributed to the simplification of the prediction task in DM, which focuses on a single class. By narrowing the scope of prediction, DM streamlines the learning process. Additionally, DM benefits from a simpler decision boundary in relation to the class predictions made by function α, allowing for more distinct and accurate predictions compared to MC, which assigns probabilities to multiple adjacent cells.

Recently, the authors conducted a more detailed investigation of the studied problem, exploring other ML tech-

niques through a series of experiments. This allowed for an evaluation of the performance of the algorithm in different configurations. Additionally, the algorithm was tested for scalability. The results have been organized into a paper and are planned to be submitted for publication in the near future.

REFERENCES

[1] J. Roux and J. V. Vuuren, "Real-time threat evaluation in a ground based air defence environment," *ORiON*, vol. 24, pp. 75–101, 6 2008. [Online]. Available: http://orion.journals.ac.za/pub/article/view/60

[2] U. J. C. of Staff, "Jp 3-01: Countering air and missile threats," *Joint Publication 3-01*, 2017.

[3] J. Roux and J. V. Vuuren, "Threat evaluation and weapon assignment decision support: A review of the state of the art," *ORiON*, vol. 23, pp. 151–187, 12 2007. [Online]. Available: http://orion.journals.ac.za/pub/article/view/54

[4] Y. Cao, Y. X. Kou, A. Xu, and Z. F. Xi, "Target threat assessment in air combat based on improved glowworm swarm optimization and elm neural network," *International Journal of Aerospace Engineering*, vol. 2021, 2021.

[5] H. Lee, B. J. Choi, C. O. Kim, J. S. Kim, and J. E. Kim, "Threat evaluation of enemy air fighters via neural network-based markov chain modeling," *Knowledge-Based Systems*, vol. 116, pp. 49–57, 1 2017.

[6] A. Kline, D. Ahner, and R. Hill, "The weapon-target assignment problem," pp. 226–236, 5 2019.

[7] T. Zhou, M. Chen, Y. Wang, J. He, and C. Yang, "Information entropy-based intention prediction of aerial targets under uncertain and incomplete information," *Entropy*, vol. 22, 3 2020.

[8] Y. Gao, D. S. Li, and H. Zhong, "A novel target threat assessment method based on three-way decisions under intuitionistic fuzzy multi-attribute decision making environment," *Engineering Applications of Artificial Intelligence*, vol. 87, 2020.

[9] L. Yue, R. Yang, J. Zuo, H. Luo, and Q. Li, "Air target threat assessment based on improved moth flame optimization-gray neural network model," *Mathematical Problems in Engineering*, vol. 2019, 2019.

[10] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, 1958.

[11] J. Roy, S. Paradis, and M. Allouche, "Threat evaluation for impact assessment in situation analysis systems," in *Signal Processing, Sensor Fusion, and Target Recognition XI*, I. Kadar, Ed., vol. 4729, 7 2002, pp. 329–341. [Online]. Available: doi:10.1117/12.477618

[12] D. H. Hong, J. H. Yoo, S. C. Shin, S. H. Kim, and Y. W. Park, "A threat evaluation method on the air track in short range air defense systems," in *Proc. Fall Conference on Korean Institute of Communication and Information Sciences*, 2011, pp. 414–415.

[13] L. Jianwei, T. Songjie, G. Qi, and C. Yanbin, "Model of large area air-defence operation efficiency evaluation based on lanchester equation," *Command Control & Simulation, vol. A247*, pp. 11–13, 1 2006.

[14] W. Wei, S. Hongquan, W. Lei, and Z. Jianhua, "Research on formation aerial target threat assessment based on risky multi-attribute decision-making," *Ship Science and Technology*, vol. 36, pp. 146–149, 9 2014.

[15] R. Zhao, F. Yang, L. Ji, and Y. Bai, "Dynamic air target threat assessment based on interval-valued intuitionistic fuzzy sets, game theory, and evidential reasoning methodology," *Mathematical Problems in Engineering*, vol. 2021, 2021.

[16] M. Liebhaber and B. Feher, "Air threat assessment: Research, model, and display guidelines," *Pacific Science*, 2002.

[17] Y. Liang, "An approximate reasoning model for situation and threat assessment," in *Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007*, vol. 4, 2007.

[18] F. Johansson and G. Falkman, *A Comparison between Two Approaches to Threat Evaluation in an Air Defense Scenario*. Springer, 2008, vol. 5285 LNAI, pp. 110–121. [Online]. Available: doi:10.1007/978-3-540-88269-5_11

[19] X. Ximeng, Y. Rennong, and Y. Yang, "Threat assessment in air combat based on elm neural network," in *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*. IEEE, 3 2019, pp. 114–120. [Online]. Available: doi:10.1109/ICAICA.2019.8873461

[20] M. Guanglei, Z. Runnan, W. Biao, Z. Mingzhe, W. Yu, and L. Xiao, "Target tactical intention recognition in multiaircraft cooperative air combat," *International Journal of Aerospace Engineering*, vol. 2021, 2021.

[21] P. Yu, G. Liu, W. Wei, and Z. Tang, "Research on airspace target threat assessment model in sea battlefield based on ga-rbf," in *2019 IEEE 2nd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*. IEEE, 11 2019, pp. 428–432. [Online]. Available: doi:10.1109/AUTEEE48671.2019.9033371

[22] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, 1990.

[23] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, 1965.

[24] H. Levene, "Robust tests for equality of variances," *Contributions to probability and statistics: Essays in Honor of Harold Hotelling*, vol. 69, 1960.

[25] W. H. Kruskal and W. A. Wallis, "Use of ranks in one-criterion variance analysis," *Journal of the American Statistical Association*, vol. 47, 1952.