Electromagnetic situation analysis and judgment based on deep learning

Yuntian Feng | Bing Li | Qibin Zheng | Dezheng Wang | Xiong Xu | Rongqing Zhang

1 State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System (CEMEE), Luoyang, China
2 School of Software Engineering, Tongji University, Shanghai, China

Correspondence
Bing Li, School of Software Engineering, Tongji University, Shanghai 201804, China.
Email: lizi@tongji.edu.cn

Abstract
The electromagnetic situation, which can promote the abilities of understanding and decision-making for the battlefield, has attracted significant interest recently in information-based warfare. This paper investigates the deep learning-based electromagnetic situation analysis and judgment in a complicated battlefield environment. To comprehensively simulate the two-sided battling process, a turn-based confrontation strategy is proposed, and an electromagnetic situation analysis and judgment model are then designed based on the AlphaGo Zero algorithm to achieve efficient situation analysis and decision-making. In addition, an electromagnetic situation-based attack-defense platform is developed to realize and evaluate this designed model. Simulation results demonstrate that this designed model achieves significant performance in electromagnetic situation analysis and judgment compared with the Monte Carlo Tree Search based baseline.

1 INTRODUCTION

Modern warfare has gradually changed from mechanized warfare to information-based warfare [1, 2]. The outcome of a war depends on the mastery and application of information. The complex electromagnetic situation is an important feature of the informationized battlefield and also a key target of the confrontation between the two sides of the war under the informationized condition. Therefore, it is urgent to establish a more effective and accurate representation and calculation model of the electromagnetic situation. In addition, the battlefield situation contains diversified information elements. The comprehensive analysis of the battlefield is inseparable from the confrontation judgment and evolution prediction of electromagnetic situation.

The electromagnetic situation has the characteristics of extensiveness, denseness, dynamism, antagonism, and relativity. The main parameters of the electromagnetic situation include signal density, signal strength, and signal type, which are distributed in four domains, namely time domain, frequency domain, space domain, and energy domain. In the modern informationized battlefield environment, electromagnetic radiation changes in the time domain, electromagnetic signal carrier frequencies overlap in the frequency domain, electromagnetic propagation intersects in the space domain, and the intensity of electromagnetic radiation fluctuates in the energy domain. Accurately perceiving and analysing complex electromagnetic situations can enable commanders to quickly and intuitively understand the battlefield, and further provide a reliable guidance for battle command decision-making as well as the use of radar and communication equipment.

Research on the electromagnetic environment is mainly based on traditional methods. In [3], the zero-sum game was adopted to model the electromagnetic confrontation between the two sides of the war in the case of manned aerial vehicles and unmanned aerial vehicles. Considering the judgment of the...
complex electromagnetic situation, in [4], an equivalent electromagnetic situation based on meta-elements was constructed, which helps to test and analyze the effect of electromagnetic situation on the electronic information system. A sensing and calculation method of quasi-static electromagnetic field was introduced in [5], which can obtain the transient electromagnetic response on the surface of a uniform half-space. In [6], in order to improve the efficiency and accuracy, the ray tracing algorithm was considered to model and calculate the electromagnetic situation. The ray tracing algorithm is an electromagnetic field strength prediction algorithm based on geometrical optics (GO) and geometric uniform theory of diffraction (UTD). A set of parallel systems is constructed to complete the accelerated simulation of complex electromagnetic situation in [7]. The parallel calculation method with multiple child nodes and the acceleration algorithm of ray tracing are combined in this system. In [8], the propagation of electromagnetic signal is calculated through dyadic Green's function (DGF). Applying the principles of human-machine Situation Awareness (SA) to the research of Electronic Warfare (EW), the research in [9] combined Endsley SA mode with the extensions of electromagnetic situation ontology. Most previous studies focus on a certain key technology in the electromagnetic situation, and cannot fully reflect the elements and correlation characteristics of the electromagnetic situation in various domains. The problem of generating a systematic battlefield electromagnetic situation has not been studied in the existing literature.

Due to high complexity, traditional electromagnetic theory-based solutions consume a lot of computing resources, which makes it difficult to obtain real-time results and cannot provide the latest electromagnetic situation data for the simulation system. On the contrary, machine learning (ML)-based modelling can reduce the calculation time and provide accurate data for the simulation system in real-time [10]. ML includes two mainstream technologies, among which deep learning (DL) is suitable for processing big data and reinforcement learning (RL) can obtain new data through the exploration of environment.

DL, as the core of artificial intelligence (AI), has demonstrated outstanding advantages. With the continuous improvement of the computers’ data processing capabilities, DL has been widely used in various scenarios, such as picture processing and natural language processing (NLP) [11]. DL was compared with traditional ML-based methods in [12]. Through the research and analysis of the network structure, advanced technical solutions of deep learning in image recognition and classification are proposed. Authors in [13] proposed a support vector machine (SVM)-based DL algorithm to monitor port scan attempts and ensure the security of network communication. In [14], convolutional neural network (CNN) was applied to realize large-scale data management and applications scheduling among IoT devices. In [15], high-precision pedestrian monitoring based on DL is realized, which can guarantee recognition speed at the same time. In the experiment of sentiment analysis on Twitter data, performance of two different machine learning methods, CNN and recurrent neural networks (RNN), were compared [16]. It can be concluded that CNN are particularly outstanding in the field of image processing, and RNN are more suitable for NLP tasks. In [17], a distributed intelligent surveillance system was built based on edge computing and cloud computing. Network bandwidth and cloud computing costs are significantly reduced through the usage of DL.

Different from other ML paradigms, such as supervised learning and unsupervised learning, RL does not require regular training samples and labels, and can achieve the purpose of learning by rewarding and punishing. RL is a method for adaptive control of non-linear systems, and it emphasizes how algorithms act on the environment. Specifically, reinforcement learning gradually forms expectations for the given reward or punishment stimulus, and eventually produces habitual behaviours that maximize benefits [18]. Derived from the statistical simulation in mathematical theory, the Monte Carlo method is a cutting-edge method in the field of RL. In [19], a scheme for uncertainty evaluation based on the Monte Carlo method was proposed, which can reduce the analysis workload of complex or non-linear models. In [20], a bounded Monte Carlo method was designed, which can improve the calculation accuracy of Monte Carlo method with the help of upper and lower bounds. In the case of multiple jamming signals, authors in [21] employed Monte Carlo method to analyze the radar monitoring performance under the interference environment. In [22], a simulation framework based on Monte Carlo method was proposed to help customers budget the cost of cloud computing services.

As the integration and extension of DL and RL, deep reinforcement learning (DRL) has attracted a significant amount of attention in data dimensionality reduction, wireless communications, and autonomous learning [23]. DRL algorithms have revolutionized the research of AI, and have taken a crucial step towards autonomous systems with a higher degree of intelligence [24]. In the antagonistic game scenario, the AlphaGo algorithm has great advantages among numerous algorithms of DRL [25]. AlphaGo, developed by Google DeepMind, provides an engineering solution to complex intelligent problems [26]. Based on the AlphaGo algorithm, the neural network tree search was enhanced, and then the AlphaGo Zero algorithm with better performance came up [27]. The basic principle of AlphaGo Zero algorithm is to combine the neural network of DL with the Monte Carlo method of RL. The specific mechanism is to train the deep convolutional neural networks (DCNN) with RL [28, 29]. Different from AlphaGo Fan and AlphaGo Lee, AlphaGo Zero algorithm can enhance its performance in Go through complete independent exploration and learning without any supervision or artificial data [27].

The confrontation judgment and evolution analysis of electromagnetic situation can be equated to the game of Go essentially. Inspired by the above discussions, in this article, we employ the AlphaGo Zero algorithm to address the electromagnetic situation analysis and judgment. The main contributions of this paper can be summarized as follows.

- To the best of the authors’ knowledge, this is the first attempt to investigate the confrontation judgment and evolution analysis of electromagnetic situation. Through the analysis and induction of the complex electromagnetic situation in the
battlefield, we design an electromagnetic situation calculation model, which describes the battle objects in the battlefield environment and their electromagnetic characteristics.

- Moreover, we design a turn-based confrontation strategy to simulate the battling process, based on which the design of the algorithm and simulation platform can be carried out. Employing AlphaGo Zero algorithm, we design an electromagnetic situation analysis and judgment model, which can realize situation analysis and decision-making in the electromagnetic battlefield through autonomous learning.
- Finally, for the analysis and judgment model, we developed an electromagnetic situation analysis and judgment simulation platform to realize the model in a visual and operable way. Simulation results verify the efficiency of our design in significantly improving the win rate in such a battlefield with electromagnetic confrontation, compared with the Monte Carlo Tree Search (MCTS)-based baseline.

The rest of this paper is organized as follows: In Section 2, the system model under the electromagnetic situation scenario is introduced. In Section 3, we present a analysis and judgment model based on the AlphaGo Zero algorithm. In Section 4, a simulation platform is set up to simulate the battle scene and demonstrates the numerical results of the designed model. Finally, Section 5 concludes the paper.

2 | SYSTEM MODEL

2.1 | System model

As illustrated in Figure 1, we consider a battlefield with the electromagnetic confrontation between two sides, called the red army and blue army. Each army consists of several battle groups, shown as Red Groups and Blue Groups, which can be further divided into multiple types of battle units, such as vehicles and airplanes. Battle units are the basic element in electromagnetic confrontation.

Battle groups with different functions include different battle units. As the detail of Red Group-2 shown in Figure 2, several adjacent battle units can form a battle subgroup, such as Missile Base, Airport, Defense Formation, and Defense System. For example, the battle subgroup Missile Base consists of two Vehicle-3. For the sake of confidentiality, different types of vehicles are represented as Vehicle-X, \(X \in \{1, \ldots, 8\}\), and different types of the airplanes are represented as Airplane-Y, \(Y \in \{1, \ldots, 4\}\). Some battle units, such as Airplane-3, Airplane-4, and Vehicle-3, can carry missiles and have attack capabilities. Different types of the missile are simplified as Missile-Z, \(Z \in \{1, \ldots, 6\}\). A battle group can be composed of battle units and battle subgroups.

The electromagnetic situation of the two armies can be obtained by the feature recognition of their battle units. In the analyzed electromagnetic situation, the two armies use a designed turn-based confrontation strategy to confront each other, attacking one battle unit or battle subgroup of each other at a time. In particular, when an Airport is destroyed, we think all the Airplanes in it are also incapacitated. Therefore, battle units and battle subgroups are all targets that can be attacked, collectively referred to as battle equipment. The confrontation ended within a limited number of rounds, and the result of the electromagnetic situation is finally estimated.
2.2 Feature recognition

In actual confrontation scenarios, battle units will send out various electromagnetic signals, such as radar data, radar reconnaissance data, communication reconnaissance data, and spectrum data. The electromagnetic signals emitted by different types of battle units have many differences, such as operating frequency, pulse repetition frequency, and pulse width. We detect and collect these signals through radar detection. Then feature extraction can be carried out by corresponding analysis, such as wavelet analysis, time domain analysis, and frequency domain analysis. Finally, the corresponding battle units are identified according to the specific signal features.

All possible battle units and their attributes are stored in our database. Battle units have the following attributes to describe and quantify their features:

- **Value**: Battle groups, battle units, and missiles have their own value. The value of the battle group and the army is equal to the combined value of the battle units they contain. The value of the two armies reflects the electromagnetic situation between them.
- **Object**: Some battle units are equipped with several types of missiles, so they have the ability to attack the enemy battle equipment. Different types of missiles can attack some types of enemy battle equipment. For example, Missile-1 can attack Airport and Vehicle-1, but not Airplane-1. The Object contains all targets that can be attacked by the missiles on the battle unit.
- **Attack value**: Missile attack will reduce the value of the Objects. The damage value of the same type of missile attacking different equipment is different, and the damage value of different missiles attacking the same equipment is also different.

2.3 Turn-based confrontation

As shown in Figure 3, the form of electromagnetic confrontation between the two armies is turn-based battle. Each round is attacked by one army. We assume that the attacker has $M$ missiles, denoted by $m_1, m_2, ..., m_M$. The attacked army has $E$ battle objects containing battle units and battle subgroups, denoted by $e_1, e_2, ..., e_E$. A specific missile can only attack a specific object, causing damage to the object's value, which is expressed as $d(m_i, e_j)$. The attacker determine a missile $m_i$ to attack an enemy battle objects $e_j$, denoted as attack strategy $m_i \rightarrow e_j$. Then, the value of the attacked object is reduced to $v(e_j) = v(e_j) - d(m_i, e_j)$. When the attack is complete, switch the roles of the attacker and the attacked army to start the next round. Keep looping until one army wins or the two armies tie. The battle ends when the number of battle rounds reaches the limit or when both armies cannot attack. When the battle ends, the army with the higher remaining value wins. If the remaining values are the same, the two armies will tie.

Through the above modelling of the electromagnetic confrontation between two armies, we find that the turn-based confrontation process is very similar to the game of Go. Therefore, in Section 3, AlphaGo Zero algorithm is applied to design an analysis and judgment model in the electromagnetic confrontation scenario.

3 ELECTROMAGNETIC SITUATION ANALYSIS AND JUDGMENT MODEL BASED ON ALPHAGO ZERO

This electromagnetic situation analysis and judgment model is employed to select the next attack action $a$ (i.e. using the missile $m_i$ to attack the opponent’s equipment $e_j$, to maximize the probability of the attacker’s final victory), which is calculated according to the current round of the battle state $s$.

This section gives a detailed introduction to the electromagnetic countermeasure analysis and judgment model based on AlphaGo Zero algorithm. First, the state definition related to the model algorithm is given, and then the composition of the algorithm and the specific details of each part are introduced.

3.1 Battle state definition

The input of the analysis and judgment model consists of the state of two armies and the information indicating the current attacker.
An army’s state is stored in a state matrix, which is updated every round. We assume that there are $K_1$ types of missiles and $K_2$ types of battle equipment. $K_1$ missile types are numbered from 0 to $K_1 - 1$, and $K_2$ equipment types are numbered from $K_1$ to $K_1 + K_2 - 1$. The total amount of $K_2$ types of battle equipment is $N$. A state matrix with 3 rows and $T$, $T = K_1 + N$, columns can be built to describe an army.

For example, Table 1 is a state matrix of an army when $T = 100, K_1 = 6, K_2 = 12$. The first $K_1$ columns contain the information of the $K_1$ types of missile, such as quantity and total value, and the last $T - K_1$ columns describe the information of each battle equipment, such as the quantity and the residual value of the device. Specifically, the first row of the state matrix indicates the number of equipment. The first $K_1$ columns show the number of missiles in this column, and the last $T - K_1$ columns show whether there exists equipment, while 1 means one equipment, and 0 means no equipment. The second row represents the type number of each column. The third row of the state matrix represents the residual value corresponding to the equipment and missile of each column. The first $K_1$ columns represent certain types of missiles, and the residual value equals the number of missiles in this column multiply the value of a single missile. For each attack, the number of missiles is reduced by one, and the corresponding residual value is also reduced by the value of a unit missile. The last $T - K_1$ columns indicate the residual value of the equipment. For example, if the equipment in a column is Airport, then the initial residual value of the column is the value of a single Airport, which is $580$. Thus, the missile Missile-1 attack damages 280, and the new value of the column is $300$.

In addition to the state matrices, the current battle situation also contains information indicating the current attacker. Assuming that one army is specified as 0 and the other is 1, the current state of the battle is $s$, $s = \{s_0, s_1, \ldots, s_T\}$. The current attacker in state $s$ can take several kinds of attack actions $a$, $a \in \{a_1, \ldots, a_n\}$. Each edge represents one attack action, and each node $s$ can be connected to several edges. For example, node $s_0$ corresponds to $n$ edges $a_1, \ldots, a_n$. The battle state $s$ changes to the new battle state $s'$ after the attack action $a$, and $s'$ is a child node of node $s$. After such a round of attacks, the identities of the attacker and the defender are exchanged, and a new attacker initiates a new attack action. Similarly, the new battle state $s'$ is also used as a node to connect several child nodes through several edges.

In addition to $s$ and $a$, Monte Carlo tree also saves 4 additional variables on each edge, which are used to guide the current attacker to choose the edge with the optimal attack among all edges. The 4 variables are shown in Table 2.

The attack strategy $\pi$ is used to describe the probability distribution of the attack action $a$ in the battle state $s$, in which the most frequently taken attack action is considered the most valuable attack action, that is, the optimal attack. Assuming that the current state of the battle is $s_t$, the attack strategy $\pi$ is defined in the process of MCTS, some battle situations that are not in the tree may be encountered. This requires a neural network to output a simulation strategy to improve the MCTS.

### 3.3 Choice of attack strategy

As mentioned above, we obtain the attack strategy of the current attacker by MCTS. The following subsection introduces the construction of Monte Carlo tree in the confrontation scenario, and the search algorithm for acquiring the attack strategy.

#### 3.3.1 Monte Carlo tree construction

In the confrontation scenario, the structure of Monte Carlo tree is shown in the left half of Figure 4. Each node represents a battle state $s$, $s \in \{s_0, s_1, \ldots, s_T\}$. The current attacker in state $s$ can take several kinds of attack actions $a$, $a \in \{a_1, \ldots, a_n\}$. Each edge represents one attack action, and each node $s$ can be connected to several edges. For example, node $s_0$ corresponds to $n$ edges $a_1, \ldots, a_n$. The battle state $s$ changes to the new battle state $s'$ after the attack action $a$, and $s'$ is a child node of node $s$. After such a round of attacks, the identities of the attacker and the defender are exchanged, and a new attacker initiates a new attack action. Similarly, the new battle state $s'$ is also used as a node to connect several child nodes through several edges.

In addition to $s$ and $a$, Monte Carlo tree also saves 4 additional variables on each edge, which are used to guide the current attacker to choose the edge with the optimal attack among all edges. The 4 variables are shown in Table 2.

The attack strategy $\pi$ is used to describe the probability distribution of the attack action $a$ in the battle state $s$, in which the most frequently taken attack action is considered the most valuable attack action, that is, the optimal attack. Assuming that the current state of the battle is $s_t$, the attack strategy $\pi$ is defined.

### 3.2 The composition of the analysis and judgment model

We apply the core algorithm of AlphaGo Zero [27] to design the analysis and judgment model in electromagnetic confrontation, as shown in Figure 4, which is mainly composed of MCTS structure and neural network. The MCTS obtains the attack strategy of the current attacker through search and calculation. However,
TABLE 2 Additional variables of each edge

| Variable | Definition |
|----------|------------|
| $N(s,a)$ | Number of visits of edge $a$ |
| $V(s,a)$ | Total action value of edge $a$ |
| $\overline{V}(s,a)$ | Average action value |
| $P(s,a)$ | Prior probability of taking action $a$ in state $s$ |

as follows [27]

$$\pi(a|s_0) = \frac{N(s_0, a)^{\tau}}{\sum_b N(s_0, b)^{\tau}},$$

(1)

where the set of all possible attack actions in state $s_0$ is denoted as $\{b\}$. $\tau$ is the temperature constant, used to control the degree of exploration. When $\tau$ is large, the probability of all attack actions is not much different, and it tends to explore all possible attack actions. When $\tau$ is small, it tends to select the most frequently visited attack action.

The attack strategy $\pi$ is calculated by $s_0$ and $a$. Thus, the optimal attack, denoted as $a_{0}$, selected by the current attacker in the battle situation $s_0$ is given by

$$a_{0} = \arg\max_{a} \pi(a|s_0),$$

(2)

3.3.2 Search algorithm

The value of the attack strategy $\pi$ is calculated based on the values of the variables on the edge, $N(s_0, a)$, $V(s_0, a)$, and $\overline{V}(s_0, a)$. And these values are calculated by downwarding a fixed number of search algorithms on node $s_0$, denoted as $t_{\text{search}}$.

The search algorithm consists of 3 steps.

(1) Selection: Starting from the root node $s_0$ and continuing downwards, each downward operation selects an attack action from all possible attack actions of the current node $s_t$. The selected attack action $a_t$ is given as follows [22]

$$a_t = \arg\max_{a} \left( \overline{V}(s_t, a) + U(s_t, a) \right),$$

(3)

where the higher $\overline{V}(s_t, a)$ the greater the reward for taking the attack action $a$ under the current status node, but when the number of simulations of the attack action $a$ is relatively small, the average action value $\overline{V}(s_t, a)$ is not high, resulting in a not being selected when searching, but this does not mean that taking the attack action is not a good choice, so the second parameter $U(s_t, a)$ is added prevent this problem. $U(s_t, a)$ is defined by a variant of PUCT [30], which is given by

$$U(s_t, a) = c_{\text{expl}} P(s_t, a) \sqrt{\frac{\sum_b N(s_t, b)}{1 + N(s_t, a)}},$$

(4)

where $c_{\text{expl}}$ is an exploration constant, which specifies the degree of exploration of the search algorithm. The larger the $c_{\text{expl}}$, the more likely it is to choose unexplored situations. The selection operation continues until encounters the leaf nodes representing that the battle is over or not.

(2) Extension: When encountering a leaf node that representing the battle is not over, it is necessary to search downward through expansion. The neural network takes the current state of the leaf node $s$ as input, and obtains the probability $p$ and action value $v$ of all possible attack actions. Based on the output of the neural network, it is possible to construct the downwardly connected edges of the leaf nodes and the subsequent connected child nodes of the edges. The 4 variable values of each edge are set as

$$N(s,a) = 0,$$

$$V(s,a) = 0,$$

$$\overline{V}(s,a) = 0,$$

$$P(s,a) = p.$$  \hspace{1cm} (5)

(3) Backtracking: When the end node is encountered downwards or the leaf nodes are expanded, the ancestor nodes need to be updated in turn, and the updated content is as

$$N(s_t, a_t) = N(s_t, a_t) + 1,$$

$$V(s_t, a_t) = V(s_t, a_t) + v',$$

$$\overline{V}(s_t, a_t) = \frac{V(s_t, a_t)}{N(s_t, a_t)},$$

(6)

where $v'$ is a variable about $v$. If the number of backtracking times is $t_{\text{back}}$, then

$$v' = (-1)^{t_{\text{back}}} \cdot v,$$  \hspace{1cm} (7)

where the range of action value $v$ is $[-1,1]$, where $[-1,0]$ represents the final loss, $(0,1]$ represents the final victory. The two adjacent nodes of the MCTS tree represent the two different sides’ perspective on the battlefield status and attack strategy. For a leaf node performing an expansion operation, $t_{\text{back}}$ is equal to 0, and its corresponding $v'$ is equal to $v$. Moreover, for the parent node of the leaf node, $t_{\text{back}}$ is equal to 1, and its corresponding $v'$ is equal to $-v$. For the leaf nodes where the battle is over, the action value $v = 1$ if win, $v = -1$ if losing, and $v = 0$ if tie.

3.4 Prediction of unknown battle situation

In the previous extension step, when the MCTS encounters an unknown battle state $s$, the relevant value of the attack action including the probability distribution $p$, and value $v$ of the attack action will be predicted through the deep neural network. The structure of the deep neural network is shown in the right half of Figure 4.
In the previous expansion step, when MCTS encountered an unknown combat state \( s \), the relevant values of the attack action include the attack distribution and the attack value will be predicted by the deep neural network.

The deep neural network contains common network layers such as Convolution, Flatten, and Dense. The neural network has one input value and two output values, which can also be expressed as

\[
\langle p, v \rangle = f(\theta, s)
\]

(8)

where \( s \) is the input battle state, \( \theta \) is the weight parameter, and \( p \) and \( v \) are output results of the neural network.

- \( p \): The current attacker’s attack strategy prediction in the battle state \( s \), that is, the probability corresponding to all possible attack actions.
- \( v \): The current attacker’s action value or outcome prediction in the battle state \( s \). \( v \) is a real number, and the range is \([-1, 1]\).

### 3.5 Training optimization of neural network model

The initial neural network model will only output random results, and cannot help the analysis and judgment of attack actions and electromagnetic situation. Therefore, training is needed to continuously improve the performance of the model. The training optimization process is shown in Figure 5, which mainly divided into three phases, that is, the self-play phase, the neural network training phase, and the neural network evaluation phase.

The purpose of the self-play phase is to generate a large amount of red-blue electromagnetic confrontation data for training neural networks. The generation of confrontation data is completed by the form of self-play of the analysis model. In each new round of red-blue electromagnetic confrontation, the state of the two armies is randomly generated. Then, based on the initial battle state, the two-sided battling process is simulated. In the battle state \( s \), the next attack strategy \( \pi \) of the attack initiator is obtained by the MCTS algorithm mentioned in Section 3.3.2. When each round of red-blue electromagnetic confrontation is over, the outcome of the round \( z \) can be obtained. In the outcome, 1 means victory, −1 means defeat, and 0 means tie. After the multi-round red-blue electromagnetic confrontation, a large amount of sample data \((s, \pi, z)\) can be simulated, which can be used as training data for neural networks.

In the neural network training phase, the neural network is trained using the sample data \((s, \pi, z)\) obtained in the self-play stage, so that the output result \( p \) and \( v \) are close to the sample data \( \pi \) and \( z \). The definition of the loss function \( L \) is given as follows [27]

\[
L = (z - v)^2 - \pi^T \log(p) + c \| \theta \|^2, \tag{9}
\]

where \((z - v)^2\) represents the error between the outcome \( v \) predicted by the neural network and the real outcome \( z \). \(-\pi^T \log(P)\) represents the error between the output strategy \( p \) of the neural network and the output strategy \( \pi \) of the MCTS algorithm. \( c \| \theta \|^2 \) is the L2 regularization term used to prevent the neural network from overfitting, where \( c \) is the weight parameter that controls the size of the term, and \( \theta \) is all the network weight parameters of the neural network.

In the neural network evaluation phase, the purpose is to save the analysis and judgment model with the best performance obtained in the training phase. Specifically, our trained analysis and judgment model is confronted with the benchmark model for \( t_{\text{test}} \) rounds. If the win rate of the analysis and judgment model is higher than before, it means that the performance has been improved, and it should be saved. After completing all the training, an analysis and judgment model with the best performance is obtained.

### 4 | SIMULATION

#### 4.1 Simulation platform construction

A simulation platform is built to simulate the two-sided battling process, which can specify the equipment configuration and confrontation parameters of two armies. The simulation interface of the platform is shown in Figure 6. The simulation interface mainly contains five areas, including Import Button, Simulation View, Simulation Setting, Information Detail, and Result Output, which are marked with red boxes and corresponding texts in the figure.

The Simulation View shows the equipment deployment of the two armies, which is divided into two parts. The upper part belongs to the red army, and the lower part belongs to the blue army. It can be seen that at the beginning, the two armies each have two battle groups and detailed equipment of the battle groups are shown in the figure. When entering the confrontation, each round of simulation will change the equipment.

**FIGURE 5** Training optimization of neural network model
The simulation interface of the platform

FIGURE 7  Sample output

The Information Detail is used to display and edit equipment information. When equipment in the Simulation View is clicked, the details of the equipment will be displayed in the Information Detail area, including Title, Type, and Subordinate. For example, Information Detail in Figure 6 shows the details of the Group-1 in the red army. That is, the title is Group-1, the type belongs to the battle group, and the subordinate is equipped with one Airplane-1 and two Airplane-2. In addition, there are two edit boxes Add and Del in the area, which can add or delete subordinate equipment.

The Import Button can import files and support the simulation platform to describe the equipment of both red and blue armies in file format. The simulation platform analyzes the content of the imported file, and then refreshes the corresponding Simulation View. Compared with manually adding or deleting equipment one by one by modifying Information Detail, file import is more convenient and can persist the changes in the case of a large number of equipment changes.

The Simulation Setting is used to initiate an electromagnetic confrontation between two armies. As shown in Figure 6, there is an input box to input the number of Simulation rounds.

Then, we click the Start button, and the confrontation process will automatically run until the specified number of rounds is finished.

Result Output is used to output the corresponding detailed confrontation results, such as the attack strategy adopted in each round, the damage value caused, and the total value of each army after each round. Figure 7 shows the output result of a certain round.

Through the configuration file, we can set the basic attributes of the confrontation model, such as the value of the object; the combat unit equipped with the missile, and the target that the missile can attack. According to the requirements of the actual scene, we set reasonable parameter configuration as the configuration file shown in schematic Figure 8. In which, valuable
TABLE 3  Red-blue confrontation model neural network structure

| Level | Network layer | Output shape | Number of parameters | Upper level connection |
|-------|---------------|--------------|----------------------|------------------------|
| 1     | Input         | (3, 100, 3)  | 0                    | /                      |
| 2     | Conv2D        | (32, 100, 3) | 896                  | 1                      |
| 3     | Conv2D        | (64, 100, 3) | 18 496               | 2                      |
| 4     | Conv2D        | (128, 100, 3)| 73 856               | 3                      |
| 5.1   | Conv2D        | (4, 100, 3)  | 516                  | 4                      |
| 6.1   | Flatten       | (1200)       | 0                    | 5.1                    |
| 7.1   | Dense         | (600)        | 720 600              | 6.1                    |
| 5.2   | Conv2D        | (2, 100, 3)  | 258                  | 4                      |
| 6.2   | Flatten       | (600)        | 0                    | 5.2                    |
| 7.2   | Dense         | (64)         | 38 464               | 6.2                    |
| 8.2   | Dense         | (1)          | 65                   | 7.2                    |

TABLE 4  Key parameters of the red-blue confrontation model

| Parameter | Meaning                                      | Value                  |
|-----------|----------------------------------------------|------------------------|
| R         | The maximum number of rounds at the end of a match | 50                     |
| K_1       | Number of missile types                      | 6                      |
| K_2       | Number of other equipment types              | 13                     |
| T         | Number of columns in the state matrix        | 100                    |
| τ         | Temperature constant                         | 1 during training and 0.001 during evaluation |
| t_search  | MCTS searches number                         | 800                    |
| c_expl    | Exploration constant                         | 5                      |
| batch_size | Number of single training data               | 512                    |
| game_batch_num | Total number of training step       | 3000                   |
| epochs    | Number of iterations per training            | 5                      |
| learn_rate| Learning rate                                | 0.002                  |
| check_frequency | Frequency of model evaluation (evaluate once every N training) | 50                     |
| c         | L_2 regular term weight                      | 0.0001                 |
| t_eval    | Evaluation of the number of matches          | 50                     |

objects denote the value of different objects. Then, battle unit equipment denotes the number of missiles carried by different battle units, which depends on their corresponding actual carrying capacity. In addition, considering that different types of missiles can attack different targets in the actual scenario, attackable of each battle unit denotes the different damage values caused by the missiles to different objects.

4.2  Analysis and judge model parameters

The neural network structure used in the experiment is shown in Table 3 [27]. The output of the neural network consists of two parts, where network layer 7.1 outputs the probability distribution of the attacker’s next attack strategy matrix, and network layer 8.2 outputs the action value of the attacker. The key training parameters are shown in Table 4.

4.3  Experimental results

A total of 3000 training are performed, in which the trained model was evaluated once after every 50 training. The performance is evaluated by confronting the trained model with a benchmark model 50 times. The more times the trained model wins, the better the performance. The benchmark model uses the MCTS model to obtain the attack strategy, and the neural network method which can improve the prediction of the unknown battle state is not adopted.

4.3.1  The training model versus random model

Figure 9 shows the the performance comparison between our training model and the random model, where the random model randomly selects the attack strategy. After repeated
adjustments and training, the best result of the training model is 39 wins out of 50. The random model carried out 10 experiments. In each experiment, the random model compete with the benchmark model 50 times. As the results shown in Table 5, the best result of random model is three wins out of 50. It can be seen that the training model has great performance improvement.

### 4.3.2 The number of wins varies the number of training

Figure 10 illustrates the number of wins of the trained model with different number of training step. First, it can be observed that when the training step is less than 750, the number of win increases rapidly as the number of training step increases. Second, as the number of training step further increases, the number of win fluctuates up and down, among which the maximum number of win is 39.

### 4.3.3 Performance evaluation comparisons for different search times

Figure 11 compares the performance evaluation of different MCTS search time. It can be seen that the performance of the model with 800 search time is significantly better than the model with 400 search time.

### 5 CONCLUSION

This paper investigates the confrontation judgment and evolution analysis of electromagnetic situation in the battlefield. To simulate the confrontation between the two armies, we designed a turn-based confrontation strategy. We also proposed an electromagnetic situation analysis and judgment model based on...
on the AlphaGo Zero algorithm. A simulation platform was built to simulate the electromagnetic confrontation between two armies. Numerical results show that our proposed model achieves significant improvement in electromagnetic situation analysis and judgment. In the design of the electromagnetic confrontation battlefield model, the current work only considers attacks between the two armies, and does not consider defensive operations. In future work, we will consider a more realistic electromagnetic confrontation battlefield by taking into account the defensive operations.

ACKNOWLEDGEMENTS

This work is supported in part by the National Natural Science Foundation of China under Grant 61901302, in part by the Natural Science Foundation of Shanghai under Grant 20ZR1462400, in part by the open research fund of State Key Laboratory of CEMEE (No. CEMEE2020K0104B), in part by the Fundamental Research Funds for the Central Universities (China), and in part by State Key Laboratory of Advanced Optical Communication Systems Networks, China.

ORCID

Bing Li https://orcid.org/0000-0001-5226-7557
Rongqing Zhang https://orcid.org/0000-0003-3774-6247

REFERENCES

1. Boguzh, VM., Brailovsky, N.N., Chirkov, D.V.: Information wars in computer communication system. In: Proceedings of the 2006 International Conference—Modern Problems of Radio Engineering, Telecommunications, and Computer Science, pp. 484–484, Lviv-Slavsko, Ukraine, 2006
2. Qin, W.: War materials scheduling management based on WebGIS. In: Proceedings of the 2010 International Conference on Computer and Communication Technologies in Agriculture Engineering, pp. 336–338, Chengdu, 12–13 June 2010
3. Gao, L., Ding, J., Lian, R.: The communication availability of the cooperative combat of manned/unmanned aerial vehicles in electromagnetic cyber confrontation environment. In: Proceedings of the 2019 IEEE International Conference on Ubiquitous Computing and Communications (UCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), pp. 112–117, Shenyang, China, 21–23 October 2019
4. Dong, J., Pu, X., Han, H.: The method of building equivalent electromagnetic environment. In: Proceedings of the 2016 Asia-Pacific International Symposium on Electromagnetic Compatibility (APEMC), vol. 1, pp. 1135–1138, Shenzhen, China (2016)
5. Fu, Q., Yan, S.: A FDTD method of quasi-stationary transient electromagnetic field. In: Proceedings of the 2007 International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications, pp. 838–841. IEEE, Piscataway, NJ (2007)
6. Li, Z., Shi, D., Gao, Y.: An acceleration algorithm of calculating electromagnetic situation based on ray tracing. In: Proceedings of the 2015 7th Asia-Pacific Conference on Environmental Electromagnetics (CEEM), pp. 350–353. IEEE, Piscataway, NJ (2015)
7. Chen, L., Shi, D., Gao, Y.: Research on the simulation system of the complex electromagnetic environment. In: Proceedings of the 2014 International Symposium on Electromagnetic Compatibility, Tokyo, pp. 45–48. IEEE, Piscataway, NJ (2014)
8. Han, F., et al.: On the computation of electromagnetic dyadic Green’s function in inhomogeneous forward and inverse scattering problems. In: Proceedings of the 2016 Progress in Electromagnetic Research Symposium (PIERS), pp. 3960–3960, Shanghai (2016)
9. Roman, V., et al.: Reusability of knowledge for deriving latent situational information in EW scenarios. In: MILCOM 2016-2016 IEEE Military Communications Conference, pp. 114–119, Baltimore, MD (2016)
10. Zhang, H., Guo, Y., Zanotto, D.: Accurate ambulatory gait analysis in walking and running using machine learning models. IEEE Trans. Neural Syst. Rehabil. Eng. 28(1), 191–202 (2020)
11. Liu, S., et al.: NLIZE: A perturbation-driven visual interrogation tool for analyzing and interpreting natural language inference models. IEEE Trans. Vis. Comput. Graphics 25(1), 651–660 (2019)
12. Dong, Y., Liang, G.: Research and discussion on image recognition and classification algorithm based on deep learning. In: Proceedings of the 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLDBDI), pp. 274–278, Taiyuan, China (2019)
13. Aksu, D., Aydin, M.: Detecting port scan attempts with comparative analysis of deep learning and support vector machine algorithms. In: Proceedings of the 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), Ankara, Turkey, 2018, pp. 77–80
14. Thompson, W., Talley, M.: Deep learning for IoT communications: Invited Presentation. In: Proceedings of the 2019 53rd Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, 2019, pp. 1–4
15. Ahmed, Z., et al.: Enhanced vulnerable pedestrian detection using deep learning. In: Proceedings of the 2019 International Conference on Communication and Signal Processing (ICSP), Chennai, India, 2019, pp. 0971–0974
16. Goularas, D., Kamin, S.: Evaluation of deep learning techniques in sentiment analysis from Twitter data. In: Proceedings of the 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), Istanbul, Turkey, 2019, pp. 12–17
17. Kaskavalci, H., Goren, S.: A deep learning based distributed smart surveillance architecture using edge and cloud computing. In: Proceedings of the 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), pp. 1–6. IEEE, Piscataway, NJ (2019)
18. Sutton, R.S., Barto, A.G., Williams, R.J.: Reinforcement learning is direct adaptive optimal control. IEEE Control Syst. Mag. 12(2), 19–22 (1992)
19. Jie, H., Huayan, C., Yun, C.: Uncertainty evaluation using Monte Carlo method with MATLAB. In: Proceedings of IEEE 2011 10th International Conference on Electronic Measurement & Instruments, Chengdu, China, 2011, pp. 282–286
20. Chen, J., Feng, L.: Using lower and upper bounds to increase the computing accuracy of Monte Carlo method. In: Proceedings of the 2010 International Conference on Computational and Information Sciences, Chengdu, China, 2010, pp. 630–633
21. Chun, W., et al.: On the radar detection analysis under the interference environment based on the monte carlo method. In: Proceedings of the 2016 11th International Symposium on Antennas, Propagation and EM Theory (ISAPE), Guilin, 2016, pp. 492–495
22. Bertot, L., Genaud, S., Gossa, J.: An overview of cloud simulation enhancement using the Monte-Carlo method. In: Proceedings of the 2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Washington, DC, 2018, pp. 386–387
23. Fu, Y., et al.: A decision-making strategy for vehicle autonomous braking in emergency via deep reinforcement learning. IEEE Trans. Veh. Technol. 69(6), 5876–5888 (2020)
24. Arulkumaran, K., et al.: Deep reinforcement learning: A brief survey. IEEE Signal Process. Mag. 34(6), 26–38 (2017)
25. Zhao, D., et al.: Special issue on deep reinforcement learning and adaptive dynamic programming. IEEE Trans. Neural Netw. Learn. Syst. 29(6), 2038–2041 (2018)
26. Wang, F., et al.: Where does AlphaGo go: From church-turing thesis to AlphaGo thesis and beyond. IEEE/CAA J. Autom. Sin. 3(2), 113–120 (2016)
27. Silver, D., et al.: Mastering the game of go without human knowledge. Nature 550(7676), 354–359 (2017)
28. Fu, M.C.: AlphaGo and Monte Carlo tree search: the simulation optimization perspective. In: 2016 Winter Simulation Conference (WSC), pp. 659–670. IEEE, Piscataway, NJ (2016)
29. Li, Z.N., et al.: AlphaGo policy network: A DCNN accelerator on FPGA. IEEE Access 8, 203039–203047 (2020)
30. Rosin, C.: Multi-armed bandits with episode context. Ann. Math. Artif. Intell. 61(3), 203–230 (2011)

How to cite this article: Feng Y, Li B, Zheng Q, Wang D, Xu X, Zhang R. Electromagnetic situation analysis and judgment based on deep learning. IET Commun. 2021;15:1455–1466.
https://doi.org/10.1049/cmu2.12161