IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 October 2023, accepted 11 November 2023, date of publication 16 November 2023,
date of current version 22 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3333666

RESEARCH ARTICLE

Network Security Situation Prediction Model Based on VMD Decomposition and DWOA Optimized BiGRU-ATTN Neural Network

SHENGCAI ZHANG, QIMING FU, AND DEZHI AN
School of Cyber Security, Gansu University of Political Science and Law, Lanzhou 730070, China

Corresponding author: Shengcai Zhang (zsc6731@gsupl.edu.cn)

This work was supported in part by the Innovation Foundation Project of Gansu Provincial Department of Education under Grant 2020A-084, Grant 2020C-29, Grant 2021CYZC-73, and Grant 2022CYZC-57; and in part by the University-Level Innovative Research Team of the Gansu University of Political Science and Law.

ABSTRACT The widespread adoption of Internet-of-Things (IoT) devices has resulted in a comprehensive transformation of human life. However, the network security challenges posed by the IoT devices have become increasingly severe, necessitating the implementation of effective security mechanisms. Network security situational awareness enables an effective network state prediction for better formulation of network security defense strategies. Existing network security situational prediction methods are typically constrained by situational sequence data, especially those sequences with a high degree of non-stationarity, leading to unstable predictions and low performance. Moreover, in real-world application scenarios, the network security situational sequences are often highly non-stationary. To address these challenges, we introduce a novel hybrid prediction model named Variational Mode Decomposition (VMD) - Dynamic Whale Optimization Algorithm (DWOA) - Bidirectional Gated Recurrent Unit (BiGRU) - Attention Mechanism (ATTN). The proposed model integrates VMD, BiGRU, ATTN, and DWOA. Initially, network security situational awareness sequences are processed using VMD to decompose them into a series of subsequences, thus reducing the non-stationarity of the original sequences. Subsequently, an enhanced DWOA optimization algorithm is introduced for tuning the hyperparameters of the BiGRU-ATTN network. Ultimately, BiGRU-ATTN is employed to predict each of these subsequences, which are then aggregated to yield the final network security situational prediction value. When compared with several existing methods on public network security datasets, the proposed VMD-DWOA-BiGRU-ATTN method demonstrated an improvement in the R^2 values ranging from 6.34% to 52.61%. These results substantiate that the model significantly enhances predictive performance.

INDEX TERMS Attention mechanism, bi-directional gated recurrent unit, dynamic whale optimization algorithm, network security situation prediction, variational mode decomposition.

I. INTRODUCTION

A. BACKGROUND

The Internet-of-Things (IoT) network is an interconnected framework that amalgamates physical entities, including sensors, the internet, and electronic devices, using a software to facilitate seamless data collection and exchange. Although the IoT paradigm promises substantial advantages and conveniences [1], it concurrently poses formidable security challenges. Intruders can exploit vulnerabilities arising from weak passwords, software flaws, or default configurations to control the IoT devices. Moreover, they can engender service unavailability by inundating IoT devices or network resources by transmitting voluminous requests or malicious codes. In practical scenarios, adversaries can manipulate the operating parameters of cardiac pacemakers, resulting in fatal consequences for patients. Similarly, they can remotely manipulate the braking system of automobiles

The associate editor coordinating the review of this manuscript and approving it for publication was Giovanni Pau.

and surreptitiously alter instructions, thereby precipitating accidents. They may also contrive ransomware to unlawfully amass profits. The adversities imposed by attacks on the IoT extend beyond substantial economic losses, encompassing potential threats to user safety and well-being. Consequently, ensuring the security of interconnected devices and data has emerged as a critical concern for individuals, organizations, and society. Although existing defensive measures include lateral isolation systems, vertical encryption authentication systems, firewalls, antivirus systems, intrusion detection, and protection systems [2], these technologies offer a passive defense. They cannot accurately discern the current security status of the IoT environment or actively defend against potential threats within the network. In pursuit of network security enhancement, researchers have proposed the network security situational awareness concept. In network security, situational awareness represents an emerging task of monitoring, analyzing, and predicting security threats and malicious activities within the network.

The cybersecurity landscape is characterized by high volatility, making it challenging to predict the cybersecurity situation [3]. This instability is often caused by emerging threats, evolving attack methods, and an increasingly complex network environment [4], [5]. All of these not only affect the reliable operation of network systems, increasing the risks faced by enterprises and individuals, but they may also lead to data breaches, financial losses, and even legal liabilities [6], [7]. It's crucial to note that cybersecurity incidents can significantly increase the operational costs of network systems [6]. On one hand, enterprises need to invest more resources in security measures, such as purchasing advanced security equipment and software or hiring specialized cybersecurity teams. On the other hand, once an attack occurs, the subsequent efforts for data recovery, system reconstruction, and legal consultation will also consume substantial financial and human resources. Therefore, accurate prediction of the cybersecurity landscape is particularly important. This not only enhances the reliable operation of network systems and reduces operational costs but also helps enterprises and individuals deploy targeted security measures more effectively, minimizing unnecessary losses [8].

B. RELATED WORKS

The importance of NSSA (Network Security Situation Awareness) is steadily rising, primarily because it allows for the assessment of the overall health and safety of the network environment. This not only includes evaluating the effectiveness of existing security measures but also involves real-time identification and tracking of abnormal behaviors and potential issues within the network, enabling timely adjustments and responses [9], such as the online cyber attack situational awareness method for power grid systems [10]. As a crucial component within the NSSA framework, network security situation prediction has become a focal point of research. The main goal of the research is to more accurately identify signs of network attacks in advance and take preventative measures promptly. This not only helps in early detection and mitigation or prevention of potential attacks but also significantly improves the accuracy and operational efficiency of Intrusion Detection Systems [11], [12]. Notably, our focus is not limited to predicting and preventing specific types of network attacks. More broadly, we are concerned with comprehensively analyzing the network environment, including but not limited to hardware security, software vulnerabilities, data integrity, and user behavior. By understanding the network's operational status from multiple angles, we can better grasp its security situation, allowing for more comprehensive and accurate predictions, thus enhancing the overall security and reliability of the entire network system.

Prediction operates on a temporal scale, so network security situation prediction is essentially about using historical sequences of network security situations for time series forecasting [13]. In the field of natural sciences, a time series is a set of variables or measured data points arranged in chronological order, recorded at specific time intervals. By applying time series models, we can mathematically reveal the inherent laws or trends governing the development of phenomena. These models are based on the interrelationships of data at different times and aim to explore their patterns of change. Utilizing historical observational data, these models can effectively make predictions and inferences [14]. Therefore, knowing only the historical data of network security posture allows us to build a time-series model to estimate future network security conditions.

Two methodological approaches are predominantly used to predict network security situations. These are mathematical statistical and machine learning methods.

Mathematical statistical methods comprehensively consider the factors that may affect the network security. This is followed by mapping these various factors into the space of network security situations using mathematical expressions. Mathematical methods commonly include the analytic hierarchy process (AHP) and the time series analysis. Li et al. used the time series analysis to predict the time series by organizing data into a time series [15], analyzing the inherent data change patterns over time, and subsequently applying these patterns to the next unobserved time series to forecast future data. Wang and Hu used the time series analysis algorithm to predict the network security situation [16], [17]. They analyzed numerous historical situation values derived from a situation assessment algorithm over a period of time to predict future network security situations. The time series analysis is a widely used mathematical statistical method with classic algorithms, such as the autoregressive model and the autoregressive integrated moving average model. However, these methods have stringent input data requirements, thereby necessitating a high degree of input sequence stability, and may not guarantee a high prediction accuracy in most scenarios. The stationarity of sequence data has an impact on predictive performance [18], [19]. Generally speaking, the more stationary the sequence, the better the predictive performance.

To address the challenges posed by the non-stationarity of sequences, some sequence prediction models employ signal decomposition algorithms to handle the sequences, including Wavelet Transform (WT), Empirical Mode Decomposition (EMD), Ensemble Empirical Mode Decomposition (EEMD), and Complementary Ensemble Empirical Mode Decomposition (CEEMD). Liu and his team [20] found that relying solely on a single model is often insufficient for accurately predicting time series with high volatility and autocorrelation. As a result, they used a combination of WT, Genetic Algorithms (GA), and Support Vector Machines (SVM) to create a hybrid prediction model. Through WT, they decomposed the original data sequence into multiple sub-signals, and then used GA to optimize the parameters of SVM, achieving quite satisfactory prediction results. Compared to WT, the EMD method can decompose a complex dataset into multiple sub-sequences of varying levels of complexity. Liu et al. [21] used an EMD-ANN hybrid model for wind speed prediction. They utilized EMD to remove noise from wind signals. Experimental results of the model demonstrated the robustness of EMD-ANN in predicting highly volatile wind signals. Yang and Wang [22] established a prediction model based on CEEMD. The experimental results of the model confirmed the effectiveness of CEEMD in improving prediction performance. Although EMD and some of its modified versions still suffer from mode mixing issues, Variational Mode Decomposition (VMD) can effectively resolve this problem. By iteratively finding the optimal solution, VMD can precisely identify the center frequency and bandwidth of each component in each round of decomposition, thus avoiding the two major shortcomings of the EMD method: end-point effects and mode mixing [23]. Abdoos [23] constructed a hybrid prediction model based on VMD. The model uses VMD to reduce data noise, employs GSO (Gravitational Search Optimization) for feature selection to eliminate data redundancy, and uses ELM (Extreme Learning Machine) as the prediction model. Experimental results on different sequence data indicated that this model has high predictive performance.

Machine learning techniques have demonstrated their efficacy in predicting the network security situation, gaining extensive utilization within this domain. The fundamental process for using machine learning methods to predict network security situations involves the usage of historical network data and related security events for model training and the input of real-time network data into the trained model to predict future network security trends. The machine learning methods commonly used to predict network security situations include support vector machines, random forests, clustering algorithms, and neural networks. Hu et al. combined MapReduce and SVM to predict the online security status. Their experimental analysis revealed that this method was more accurate and efficient than traditional methods in predicting the online security status [17]. Zhang et al. used the bootstrap method to train multiple SVMs and employed the negative correlation learning theory to increase the diversity between SVMs [24]. They then optimized the SVM using GeesePSO to relatively accurately predict network security situations.

With their rapid evolution, machine learning methods have emerged as one of the most effective techniques for time series prediction. Deep learning methods in machine learning have gained widespread adoption in various fields. A growing number of scholars have begun employing them in the realm of forecasting network security situations.

Deep learning models are better at predicting network security situations than traditional machine learning models [25], [26], [27]. One reason for this is that these models can infer intricate patterns from data, which traditional models cannot [28], [29], [30]. In addition, deep learning models employ end-to-end learning, consequently reducing the influence of human intervention during the calculation process because researchers only should provide raw data. Zhang et al. used a decision tree algorithm and long short-term memory (LSTM) to predict network security situations, yielding realistic predictions [31]. Shang et al. proposed a prediction method for network security situations based on the LSTM and XGBoost, which enhanced prediction accuracy [32]. Zhu et al. addressed the inefficiency of traditional neural networks by optimizing the LSTM using an improved Nadam algorithm, which resulted in increased training speed and prediction accuracy [33]. Yao et al. proposed a temporal convolutional network (TCN) and bidirectional (BILSTM) combination model to predict network security situations [34]. The TCN was used to extract the time series features. The BILSTM was then implemented for the situation prediction, which resulted in more precise and stable predictions. Zhao et al. proposed a novel prediction model to mitigate the impact of data noise on the prediction performance [35]. Empirical mode decomposition (EMD) was employed to denoise the data before using the LSTM to predict future situation values. The trained model significantly enhanced the prediction performance. Yin et al. used a TCN and a transformer combination model to predict network security situations [25]. The TCN was applied to extract short-term features from the sequence. This was followed by the implementation of the transformer for capturing long-term correlations, which effectively improved the prediction accuracy. RNN (Recurrent Neural Network) has been widely applied to sequence prediction models. A new improved RNN network—GRU (Gated Recurrent Unit)—has demonstrated strong nonlinear fitting capabilities in the field of prediction [36]. Moreover, the GRU network has a simpler structure and faster operation speed compared to the LSTM model. Zhu and colleagues used a VMD-BiGRU model to forecast rubber futures time series and found that BiGRU, a bidirectional neural network, has significant advantages in terms of fitting performance and trend prediction [37]. Zhao and others designed an attention-based Long Short-Term Network for cybersecurity situation prediction, using the attention mechanism to optimize the prediction model and improve its accuracy. Experimental results proved that

this method has significantly higher prediction accuracy [38]. Kong and colleagues proposed a sequence prediction model based on EMD decomposition and optimized through the attention mechanism for GRU. Since GRU treats input variables equally but different variables have different impacts on the outcome, the attention mechanism is used to compensate for this limitation in GRU, paying more attention to important features and further improving prediction accuracy [39]. The relationship between the performance of neural networks and the settings of hyperparameters is very close. They have a significant impact on the model's learning ability, speed, and performance [40].

To address the issue of finding the optimal combination of hyperparameters, an increasing number of sequence prediction models are starting to apply swarm intelligence optimization algorithms to search for the best set of hyperparameters. Li and others proposed a long-time series prediction model that uses PSO (Particle Swarm Optimization) to optimize the hyperparameters of Informer, thereby improving the accuracy of predictions. Experimental results show that the predictive performance of Informer, when enhanced with optimization algorithms, outperforms that of a standalone Informer [41]. In recent years, with the development of swarm intelligence optimization algorithms, many algorithms with excellent optimization performance have emerged. Among them, WOA has outstanding optimization capabilities. Its strong local and global search abilities have led many researchers to apply it in their respective fields [42], [43], [44], [45], [46], [47]. With the rapid development of neural networks, WOA has also been used to optimize neural network parameters [48], [49]. In the prediction field, WOA has also found many applications. Peng and others proposed the HWOA-ELM model for load prediction, utilizing HWOA to optimize the parameters of ELM (Extreme Learning Machine). The predictive results show that the ELM model optimized with HWOA has superior predictive performance [50].

C. MOTIVATION AND CONTRIBUTION

Herein, we suggest a new integrated prediction model that uses variational mode decomposition (VMD) and a dynamic whale optimization algorithm (DWOA) optimization technique for the bidirectional gated recurrent unit-attention (BiGRU–ATTN) neural network to improve the prediction precision. The model employs multi features data as the basis and decomposes the network security situation data sequence into subsequences through VMD, thereby effectively removing noise from the original data. The BiGRU model is then used for the prediction, capturing correlations in the situation sequence. The attention mechanism dynamically assigns different weights to input data to highlight the most important parts of the sequence, thereby improving the prediction accuracy of the GRU model. Subsequently, the DWOA algorithm is employed to optimize the network's hyperparameters and improve the model's prediction performance. The experimental results illustrate the superior performance of the proposed model over commonly used prediction models, substantiating its efficacy. The proposed model also produces a prediction curve that fits the true value curve of the network security situation well and performs better than several existing prediction models.

The study contributions are as follows:

The role of the VMD decomposition algorithm in network security situation prediction is explored. To the best of our knowledge, this is the first application of the VMD decomposition algorithm in network security situation prediction. The results indicate that, in cases where the network security situation sequence exhibits a high nonstationarity and is difficult to accurately predict, VMD decomposes the original sequence into multiple subsequences, effectively improving the sequence stationarity and significantly enhancing the prediction accuracy.

A new optimization algorithm, called the DWOA, is proposed. The DWOA is an improved version of the original whale optimization algorithm that incorporates three optimization strategies. Compared to traditional optimization algorithms, the DWOA exhibits a higher optimization accuracy, a faster convergence speed, and most importantly, an excellent ability to escape the local optima.

A novel network security situation prediction model, called VMD–DWOA–BiGRU–ATTN, is proposed. This model combines VMD, DWOA, and a BiGRU model with a self-attention mechanism to effectively improve the network security situation prediction performance.

Compared to other prediction models, this model exhibits smaller errors and a higher fitting degree between the predicted and true network security situation curves, suggesting that it outperforms others in terms of the network security situation prediction performance.

D. ORGANIZATION

The remainder of this paper is structured as follows: Section II provides an overview of the relevant models and VMD algorithm; Section III presents the proposed whale optimization algorithm based on the opposite learning and differential evolution strategy and collaborative improvement of nonlinear convergence factors; Section IV discusses in detail the VMD–BiGRU–ATTN network security situation prediction model based on the DWOA; Part V elaborates on the simulation experiments conducted on the DWOA and proposed model and analyzes the experimental results; and Section VI summarizes the work performed herein, along with a perspective on future research.

II. CORRELATION MODEL AND DECOMPOSITION ALGORITHM

A. VARIATIONAL MODE DECOMPOSITION

VMD is an adaptive and nonrecursive signal-processing technique that allows the estimation of the number of mode decompositions for a given sequence based on the

prevailing conditions [51]. In the subsequent search and solving process, it can match the best center frequency and limited bandwidth of each mode, thereby realizing an effective separation of the intrinsic mode components and frequency domain partitioning. Compared to empirical mode and wavelet decomposition, VMD exhibits a superior performance in restoring the original signal and possesses greater noise robustness. This method enables signal decomposition into K different frequency bands and relatively stationary subsignals, as required. The central idea of the algorithm is to establish an optimal solution to the variational problem with the following specific steps:

First, a variational problem is formulated with the assumption that the original signal S is decomposed into K components μ, guaranteeing that the decomposition sequence comprises mode components with finite bandwidth and center frequency. The goal is to keep the sum of all estimated bandwidths equal to the original signal while minimizing the sum of the estimated bandwidths for each mode. This is accomplished by constructing a constrained variational expression, as (1) depicts below:

$$\begin{cases} \min\limits_{\{u_k\},\{\omega_k\}} \left\{ \sum\limits_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \dfrac{j}{\pi t} \right) \cdot u_k(t) \right] e^{-j\omega k t} \right\|_2^2 \right\} \\ s.t. \sum\limits_{k=1}^{K} u_k = S \end{cases} \tag{1}$$

where u_k represents the kth mode component; ω_k represents the frequency center of the kth mode component; and $\delta(t)$ represents the Dirac distribution.

Next, the constrained variational expression is solved by incorporating a penalty parameter α and a Lagrange multiplier operator λ. This transformation converts the constrained variational problem into an unconstrained one, resulting in an augmented Lagrange expression (2):

$$\begin{aligned} &L\left[\{u_k(t)\},\{\omega_k\},\lambda(t)\right] \\ &= \alpha \sum\nolimits_{k=1} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) \cdot u_t(t) \right] e^{-j\omega_k t} \right\|_2^2 \\ &\quad + \left\| S(t) - \sum\nolimits_{k=1}^{K} u_k(t) \right\|_2^2 + \left\langle \lambda(t), S(t) - \sum\nolimits_{k=1}^{K} u_k(t) \right\rangle \end{aligned} \tag{2}$$

Parameters u_1, ω_2, λ^1, and n are initialized to obtain the optimal solution. n is set to an initial value of 0. A loop process is then established, where n is incremented by 1. The u_1, ω_2, and λ^1 values are updated as follows:

$$\hat{u}_k^{n+1}(\omega) = \frac{\hat{f}(\omega) - \sum_{i \neq k} \hat{u}_i(\omega) + \hat{\lambda}(\omega)/2}{1 + 2\alpha(\omega - \omega_k)^2} \tag{3}$$

$$\omega_k^{n+1} = \frac{\int_0^\infty \omega \left| \hat{u}_k(\omega) \right|^2}{\int_0^\infty \left| \hat{u}_k(\omega) \right|^2} \tag{4}$$

$$\hat{\lambda}^{n+1}(\omega) = \hat{\lambda}^n(\omega) + \tau \left(\hat{f}(\omega) - \sum\nolimits_{k=1}^{K} \hat{u}_k^{n+1}(\omega) \right) \tag{5}$$

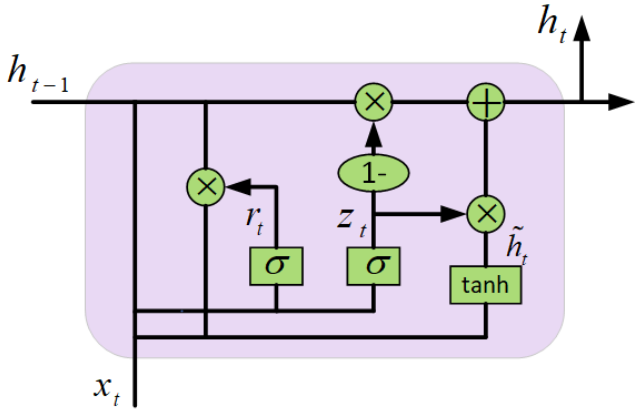

FIGURE 1. GRU structure.

B. GRU

The GRU network is a type of recurrent neural network (RNN) and an LSTM variant employed for processing the time series data and natural language processing tasks [52]. Compared to the standard RNN, the GRU network exhibits stronger modeling capabilities and a higher efficiency. It regulates the flow of the input, output, and hidden states through gate mechanisms to better capture the long-term sequence dependencies. The GRU structure is simpler than the LSTM structure, facilitating its implementation. FIGURE 1 illustrates the detailed architecture of the GRU.

The input-output architecture of the GRU network is analogous to that of a conventional RNN. Specifically, the input comprises the input at time t, denoted by x_t, and the hidden state at time t-1, represented by h_{t-1}, which incorporates pertinent information from earlier nodes. The output is the hidden node output at time t, denoted by h_t. The GRU network acquires two gating states by exploiting the previous state h_{t-1} passed down from the previous time step and the current input x_t. The GRU network employs an update gate to regulate the amount of historical information that the current state should retain from the past state and the quantity of new information it should incorporate from the candidate state, as expressed by (6).

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{6}$$

where z_t represents the update gate, and its operation is shown in the equation (7). The candidate hidden state $\tilde{h}_t$ is calculated using (8).

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t]) \tag{7}$$

$$\tilde{h}_t = \tanh(x_t W_{hx} + (r_t \odot h_{t-1}) W_{hh} + b_h) \tag{8}$$

One of the components in the GRU is the reset gate, denoted as r_t. The computation process of the reset gate is described by (9).

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t]) \tag{9}$$

The reset gate in GRU is employed to control the weighting between the current input and the previous hidden state at

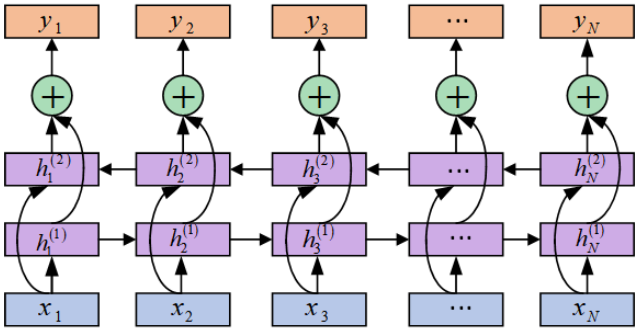

FIGURE 2. BIGRU structure.

the current time step. It combines the previous hidden state and the current input through a sigmoid function, outputting a value between 0 and 1 that indicates how much information from the previous time step should be retained. If the output value of the reset gate is close to 0, the previous hidden state will be ignored, and the current input will be processed independently. If the output value of the reset gate is close to 1, all the information from the previous time step will be retained. This mechanism enables GRU to handle long sequences more effectively and reduces the occurrence of gradient vanishing problems.

C. BIGRU

The bidirectional GRU network comprises two GRU layers that share the same input, but propagate information in opposite directions (i.e., forward and backward GRU layers).

Assume that the first layer of the bidirectional GRU network propagates information in the chronological order of time, while the second layer propagates information in the reverse order of time. The hidden layer state of BiGRU at time t is expressed in (10) to (12):

$$h_t^{(1)} = f\left(W_{hh}^{(1)} h_{t-1}^{(1)} + W_{hx}^{(1)} x_t + b_h^{(1)}\right) \tag{10}$$

$$h_t^{(2)} = f\left(W_{hh}^{(2)} h_{t-1}^{(2)} + W_{hx}^{(2)} x_t + b_h^{(2)}\right) \tag{11}$$

$$h_t = h_t^{(1)} \oplus h_t^{(2)} \tag{12}$$

where $\oplus$ is the vector concatenation operation.

FIGURE 2 depicts the bidirectional GRU structure expanded by time.

D. SELF-ATTENTION MECHANISM

The self-attention mechanism can weight and combine information from different input sequence positions based on their relationships, making it highly flexible in processing sequential data [53]. In time series prediction, attention mechanisms can improve the model's prediction accuracy by learning the relationships between different time points in the sequence. Attention mechanisms can introduce weights or attention distributions in the model to emphasize or diminish the contribution of certain time points to the prediction, thereby improving the model accuracy in predicting critical time points.

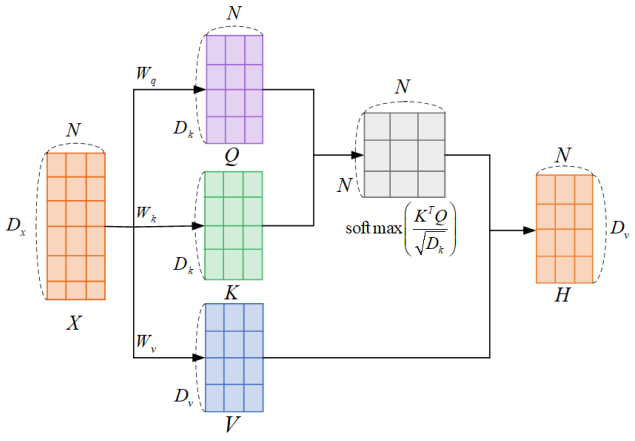

FIGURE 3. The computational process of self-attention mechanism.

The self-attention model often adopts the query–key–value (QKV) mode. FIGURE 3 illustrates its calculation process.

The specific calculation process of the self-attention model is described below assuming an input sequence $X = [x_1, \cdots, x_N] \in \mathbb{R}^{D_x \times N}$ and an output sequence $H = [h_1, \cdots, h_N] \in \mathbb{R}^{D_v \times N}$.

We first linearly map each input x_i to three distinct spaces to obtain the query vector $q_i \in \mathbb{R}^{D_k}$, key vector $k_i \in \mathbb{R}^{D_k}$, and value vector $v_i \in \mathbb{R}^{D_v}$.

For the entire input sequence X, the linear mapping process is abbreviated as follows:

$$Q = W_q X \in \Re^{D_k \times N} \tag{13}$$

$$Q = W_q X \in \Re^{D_k \times N} \tag{14}$$

$$Q = W_q X \in \Re^{D_k \times N} \tag{15}$$

(1) Matrices $W_q \in \mathbb{R}^{D_k \times D_x}$, $W_k \in \mathbb{R}^{D_k \times D_x}$, and $W_v \in \mathbb{R}^{D_v \times D_x}$ correspond to the parameter matrices used for linear mapping, while matrices $Q = [q_1, \cdots, q_N]$, $K = [k_1, \cdots, k_N]$, and $V = [v_1, \cdots, v_N]$ represent those composed of the query, key, and value vectors, respectively.

(2) The output vector h_n for each query vector $q_n \in Q$ can be obtained using (16).

$$h_n = \text{att}\left((K, V), q_n\right) = \sum_{j=1}^{N} \alpha_{nj} v_j$$

$$= \sum\nolimits_{j=1}^{N} \text{softmax}\left(s(k_j, q_n)\right) v_j \tag{16}$$

Variable α_{nj} represents the weight of the nth output vector attending to the jth input vector in the sequence, where $n, j \in [1, N]$ denotes the output and input vector positions, respectively.

III. DYNAMIC WHALE OPTIMIZATION ALGORITHM

A. WHALE OPTIMIZATION ALGORITHM

The WOA is a heuristic optimization algorithm that is based on the behavior patterns of whales. The algorithm was proposed by Mirjalili et al. in 2016 [54]. The inspiration for

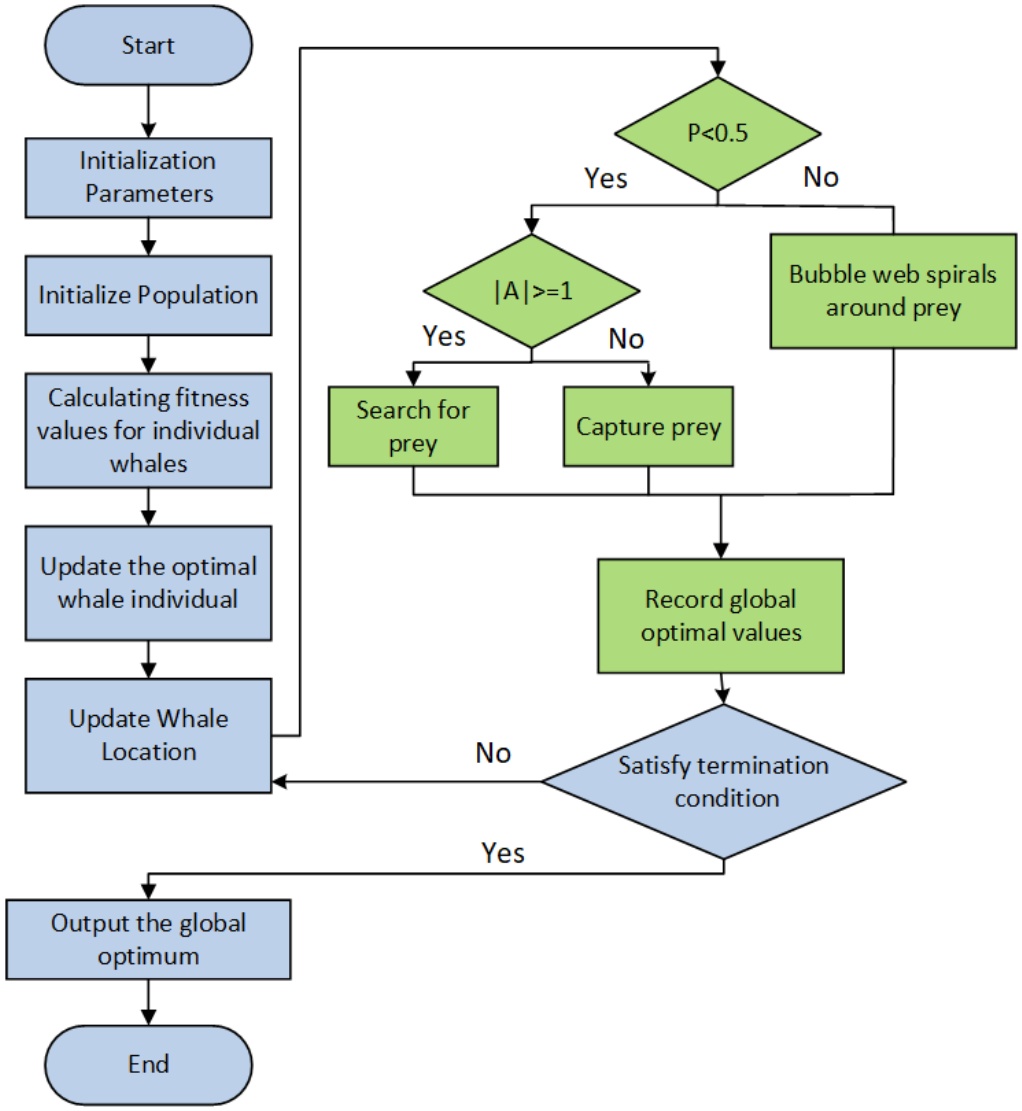

FIGURE 4. Whale optimization algorithm process.

the algorithm design comes from the collective behavior of whales, in which superior individual whales can lead others in the direction of better food sources. The basic idea of the algorithm is to treat the search space as an ocean, where each individual represents a whale. The algorithm aims to find an optimal global solution by continuously adjusting the position and velocity of the whales. The WOA is divided into three parts: 1) prey searching through a random movement; 2) surrounding the target by shrinking the search space; and 3) capturing the target through a spiral approach.

The algorithm begins by defining its relevant parameters. The iteration number is specifically represented by t, while p and l are random numbers between 0 and 1 and -1 and 1, respectively. t_{max} is the maximum iteration number. $\vec{r}_1$ and $\vec{r}_2$ are random vectors drawn from the interval [0,1]. $\vec{a}$ is a vector whose value linearly decreases from 2 to 0. (17) to (19) are used to calculate each parameter.

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{17}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{18}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{19}$$

In the WOA, whales hunt for prey using two methods. When $p < 0.5$, the whale population selects the method of contracting and surrounding the target. When $p \geq 0.5$, the whale population selects the mode of spiral hunting for prey.

1) SHRINK AND SURROUND PREY

In the encircling prey stage, the whale pod chooses either a random individual or the best individual as a reference for updating positions based on the parameter $|A|$ value. If $|A| \geq 1$, the algorithm randomly selects an individual as the reference and updates the positions using (20). If $|A| < 1$, the algorithm selects the current best individual for position updating, as illustrated in (21). The foregoing describes the encircling prey stage of the whale optimization algorithm:

$$X^{t+1} = X^t_{rand} - A \cdot \left| C \cdot X^t_{rand} - X^t \right| \tag{20}$$

$$X^{t+1} = X^t_{best} - A \cdot \left| C \cdot X^t_{best} - X^t \right| \quad (21)$$

X^t_{rand} and X^t_{best} are the positions of the random and best whale individuals in the population of generation t, respectively.

2) SPIRAL PREY

In addition to using the encircling prey technique, whales employ a spiral movement pattern to create a bubble net for hunting. The spiral hunting technique is described as follows (22):

$$X^{t+1} = X^t_{best} + \left| C \cdot X^t_{best} - X^t \right| \cdot e^{bl} \cdot cos\,(2\pi l) \quad (22)$$

where b is a constant controlling the helix shape.

FIGURE 4 illustrates the WOA flow.

B. OPPOSITION-BASED LEARNING

The initial population in swarm intelligence optimization algorithms is usually randomly generated. The optimal solution is unknown; hence, the population's random initialization increases the search space for feasible solutions. The larger search space results in longer search times and may cause the algorithm to converge to the local optima. Therefore, the algorithm performance is greatly affected by the population initialization.

The idea of the opposition-based learning method is to search for the opposing individual of each member of the population within the search space. The opposing solution replaces the original one if it performs better [55]. This approach increases the probability of getting closer to the optimal global solution by 50% compared to the initial solution and effectively improves the algorithm's optimization efficiency. The calculation process is described as follows (23):

$$x^* = a + b - x \quad (23)$$

where a is the upper boundary of the search space; b is the lower boundary; and x^* is the opposite solution. Likewise, inverse learning can be extended to higher dimensions with (24):

$$x^*_i = a_i + b_i - x_i \quad (24)$$

Assuming that $fitness\,(x)$ is the fitness function, X_i is the ith individual in the current population, and X^*_i is the reverse individual generated after reverse learning. If $fitness\left(X^*_i\right) < fitness\,(X_i)$, X_i is replaced by X^*_i and proceeds to the next population generation.

C. NONLINEAR CONVERGENCE FACTOR

Swarm intelligence optimization algorithms typically go through two stages during the optimization process: global and local search stages. Failure to coordinate these two steps may result in a slow convergence or a premature stabilization of the algorithm. The global search stage primarily aims to expand the search area, maintain the population diversity, and reduce the probability of the algorithm getting stuck in a local optimal solution. The local search stage mainly aims to perform precise searches in the search space of the population individuals, which can accelerate the convergence efficiency while improving the convergence accuracy. As a swarm intelligence optimization algorithm, the WOA undergoes these two stages, as well, thus requiring a balance between these stages. Based on its optimization principle, the WOA is known to use parameter A to balance its global and local search abilities. As shown in (17), the value of parameter A changes with the variation of the convergence factor a. Therefore, the convergence factor a value plays a crucial role in balancing the global and local search abilities.

Existing studies have demonstrated a strong correlation between the performances of the global and local search abilities of the WOA and the convergence factor denoted by a. In the classical WOA, a linearly decreases from 2 to 0 as the population evolves, consequently failing to accurately reflect the actual optimization process. A novel method for computing a to achieve a better balance between the algorithm's global and local search capabilities has been proposed and shown in (25).

$$a = 2 \times \left(\frac{t_{max}}{t_{max}}^{\frac{1}{8}} \right) \quad (25)$$

where t_{max} is the maximum number of iterations, and t is the current iteration number. The original convergence factor a linearly decreases during the iteration process. The improved convergence factor a nonlinearly decreases at a slower rate in the early iteration stage, maintaining a relatively large value for parameter A in the early algorithm iteration stage and improving the global search performance. In the later iteration stage, a nonlinearly decreases at a faster rate, enabling parameter A to maintain a small value in the later iteration stage and effectively improving the local search performance. In summary, the improved convergence factor a effectively balances the global and local search stages, fully unleashing their respective abilities.

D. DIFFERENTIAL EVOLUTION STRATEGY

Differential evolution (DE) is a population-based evolutionary algorithm that simulates the cooperative and competitive processes among individuals in a population [56]. In the DE context, the gene of each individual corresponds to a potential solution for the given problem. A mutation operation is first performed at each iteration to select the genes of one or more individuals as a basis. The results of the differential operation of different individuals are then selected to form differential genes. Finally, the base and differential genes are added to generate a new individual. With a certain probability, the mutation and original vectors for each individual are crossed to generate a new individual vector. The newly generated individual is then compared to its corresponding parent individual, and the superior one is selected and retained for the subsequent generation. The gene of the top-performing

individual within the selected population is designated as the solution upon the completion of the iterations.

1) INITIALIZATION OPERATION

The D-dimensional parameter vector X_i^t with NP randomly initialized values is expressed mathematically as follows.

$$X_i^t = \left(x_{i,1}^t, x_{i,2}^t, \cdots, x_{i,D}^t\right) \; i = 1, 2, \cdots, NP \; j = 1, 2, \cdots, D \tag{26}$$

2) MUTATION OPERATION

For each solution vector X_i^t, when a mutation operation is performed, the algorithm randomly selects three other vectors from the population and uses the difference between two of them added to one of the vectors to obtain a new solution vector as a result of the mutation operation. The specific mutation operation process is depicted in (27).

$$U_i^t = X_{r1}^t + F \cdot \left(X_{r2}^t - X_{r3}^t\right) \tag{27}$$

3) CROSS OPERATION

In the crossover operation, the algorithm transforms the mutation vector $U_i^t = \left(u_{i,1}^t, u_{i,2}^t, \cdots, u_{i,D}^t\right)$ into a new vector $V_i^t = \left(v_{i,1}^t, v_{i,2}^t, \cdots, v_{i,D}^t\right)$ according to (28).

$$v_{i,d} = \begin{cases} u_{i,d}, \, rand\,(0,1) < CR \quad or \; d = d_{rand} \\ x_{i,d}, \, rand\,(0,1) \geq CR \quad or \; d = d_{rand} \end{cases} \tag{28}$$

where the crossover probability is denoted as CR, and d_{rand} represents a random integer uniformly distributed between 1 and D. This integer guarantees the involvement of at least one dimension of the mutated vector in the crossover operation. The crossover operation involves the selection of at least one dimension of the mutated vector to replace the corresponding dimension of the original vector and obtain the final crossover vector.

4) SELECTION OPERATION

The relative fitness values of the crossover V_i^t and original X_i^t vectors are compared to determine the crossover vector selection. The selection process is governed by (29).

$$X_i^{t+1} = \begin{cases} V_i^t, & f\left(V_i^t\right) < f\left(X_i^t\right) \\ X_i^t, & f\left(V_i^t\right) \geq f\left(X_i^t\right) \end{cases} \tag{29}$$

where X_i^{t+1} is an individual in the next generation population.

FIGURE 5 demonstrates the flow of the DE strategy.

E. DWOA ALGORITHM PROCESS

An ideal WOA should ensure optimum efficiency and accuracy while avoiding falling into the local optima. A single improvement strategy cannot achieve this goal. Therefore, the DWOA employs a mixture of various improvement strategies to solve complex optimization problems. A modified initialization strategy is proposed to overcome the diminished optimization efficiency resulting from a randomly generated initial population. Furthermore, an opposition-based learning strategy is introduced to increase the population diversity, generate an initial population closer to the optimal solution, and improve the optimization efficiency. The risk of falling into the local optima is an important factor affecting the algorithm performance. The DE strategy is introduced to tackle this problem by applying mutation, crossover, and selection operations to the population in each iteration process, effectively improving the population diversity, and increasing the algorithm's probability of escaping from the local optima. The original convergence factor is improved to a nonlinear convergence factor, which fits well with the optimization process of the WOA, to better coordinate the global and local search performances. The control parameter A is dynamically adjusted by incorporating a nonlinear convergence factor to maintain higher and lower values during the early and later stages of the optimization process, respectively. This adaptive adjustment optimizes the balance between the global and local search capabilities, enhancing the likelihood of escaping the local optima and improving the overall optimization efficiency. The specific DWOA process is as follows:

Step 1: Initialize the population and the parameters: The population size, number of iterations, vector dimensions, upper and lower bounds of the search space, and various vector parameters are set during initialization.

Step 2: Update individual positions: When p is less than 0.5, and $|A|$ is less than 1, the algorithm updates the positions of each individual in the population by selecting the optimal individual using (21). If p is less than 0.5, and $|A|$ is greater than or equal to 1, the algorithm updates the positions of each individual in the population by selecting a random individual using (20). If p is greater than or equal to 0.5, the algorithm employs a bubble net selection strategy based on (22) to update the positions of the individuals in the population.

Step 3: Implement the DE strategy for the individuals generated in the current iteration.

Step 4: Repeat steps 2 and 3 until the termination condition is met.

Step 5: Satisfy the termination condition (reach the maximum number of iterations) and output the optimal individual to end the algorithm.

FIGURE 6 illustrates the DWOA flow. Algorithm 1 is the pseudo code of the DWOA.

IV. THE NETWORK SECURITY SITUATION PREDICTION MODEL BASED ON DWOA OPTIMIZATION FOR VMD-BIGRU-ATT

A. DWOA BASED HYPERPARAMETER OPTIMIZATION OF BIGRU-ATTN MODEL

Different hyperparameters may significantly affect the model's performance during the training period. The number of neurons in the network, batch size of the training data, number of training iterations, and learning rate are all critical hyperparameters. The number of neurons directly influences

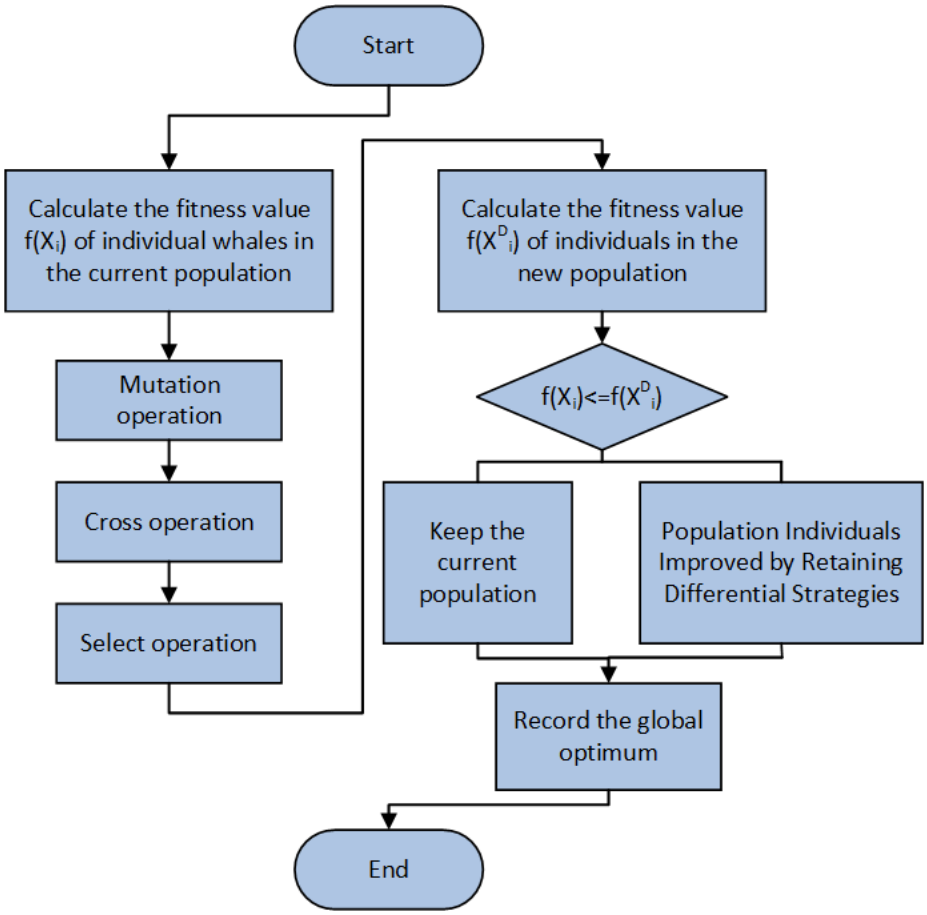

FIGURE 5. The flow of the differential evolution strategy.

the network's capacity to fit complex patterns and capture intricate relationships. The batch size and the number of training iterations play crucial roles in shaping the model's training performance. In this study, we employed the DWOA to optimize the hyperparameters and improve the model's predictive performance. The specific optimization steps are as follows:

Step 1: Initialize the population parameters and determine the number of algorithm iterations, population size, convergence factor, crossover probability, and other parameters.

Step 2: Randomly generate a four-dimensional vector. Each vector dimension is the hyperparameter for optimization.

Step 3: Use the mean square error (MSE) as the fitness function. The smaller the fitness function, the better the whale individual, that is, the parameter combination.

Step 4: Update the positions of the whale individuals and generate new ones using the DE strategy. Compare the fitness functions of the newly generated individuals and the original ones. If the fitness of the new individual is lower, it replaces the original individual in the next iteration.

Step 5: The optimization process stops when the maximum iteration limit is reached. The whale at this stage's optimum global position represents the optimal parameter combination.

B. NETWORK SECURITY SITUATION PREDICTION MODEL

This study presents a combined model that employs VMD and an improved whale optimization algorithm, called DWOA, to optimize the BiGRU–ATTN neural network. VMD decomposes the network security situation sequence into multiple modal components, thereby effectively mitigating the noise, enhancing the sequence stationarity, and reducing its complexity. The BiGRU model exhibits a robust nonlinear predictive ability and can capture long-term correlations within the sequence. The attention mechanism can weigh the significance of each time step in the sequence, enabling the model to focus more on relevant input information for prediction and reducing the interference from irrelevant information, thereby enhancing the sequence prediction accuracy. The model's prediction performance is further improved by optimizing the network hyperparameters using the DWOA.

The overall structure of VMD—DWOA–BiGRU–ATTN is shown in the FIGURE 7. The model prediction steps are as follows:

Algorithm 1 The Pseudo of DWOA

Input: *fitness* (x)-Fitness function, N-population number, max_iterations-maximum number of iterations, dim-dimension of search space, ub-upper bound of search space, lb-lower bound of search space
Output: Optimal individual

```
Initialize Whale Population x_i (i = 1, 2, ··· , n)
x_i^* = ub + lb − x_i (i = 1, 2, ..., n)
x_i ∪ x_i^* = x_j^new (i = 1, 2, ..., nj = 1, 2, ..., 2n)
fitness_j = fitness (x_j^new)
Sort x_j^new base on fitness_j
x_i ← select top n of x^new
While (t < max_iterations)
    for i in x_i
        Updated a, A, C, l, p
        if (p < 0.5  |A| < 1)
            Find the optimal individual x_best
            Update the current individual's position based on the x_best
            X^{t+1} = X_best^t − A · |C · X_best^t − X^t|
        else if (p < 0.5  |A| ≥ 1)
            Randomly select an individual x_rand
            Update the current individual's position based on x_rand
            X^{t+1} = X_rand^t − A · |C · X_rand^t − X^t|
        else if (p ≥ 0.5)
            X^{t+1} = X_best^t + |C · X_best^t − X^t| · e^{bl} · cos (2πl)
        end if
    end for
    Correcting individuals beyond the search scope
    x_i' ← x_i (i = 1, 2, ..., n) base on differential evolution strategy
    x_i ∪ x_i' = x_j (j = 1, 2, ..., 2n)
    fitness_j = fitness (x_j)
    Sort x_j base on fitness_j
    x_i ← select top n of x_j
    t = t + 1
end while
return Optimal individual
```

Step 1: Use VMD to decompose the network security situation sequence into multiple modal components.

Step 2: Feed each modal component individually into DWOA–BiGRU–ATTN for the prediction.

Step 3: Derive the actual predicted network security situation value by overlaying the predicted values of individual modal components.

FIGURE 8 illustrates the model prediction process.

V. EXPERIMENT AND DISCUSSION

This section describes our experiments and discusses the obtained results. The experiments were conducted on a system with 32 GB memory and an Intel Xeon CPU E5-2609 processor.

A. PERFORMANCE EVALUATION OF DWOA ALGORITHM

We conducted a comparative analysis with five other optimization algorithms, namely PSO, GA, SSA, WOA, and GWO, on eight benchmark test functions to validate the efficacy of the DWOA. In the experiment, we set the population size to 30, the dimensionality to 30, and the iterations to 1000. Each algorithm was independently run for 30 times. TABLE 1 lists the calculation formulas and constraints of the eight benchmark test functions.

In these test functions, the specific forms of parameters a_{ij}, a_i, and b_i in the last two fixed-dimensional test functions are shown in (30) to (32).

$$a_{ij} = \begin{bmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \cdots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \cdots & 32 & 32 & 32 \end{bmatrix} \tag{30}$$

$$a_i = \begin{bmatrix} 0.1957 & 0.1947 & 0.1735 & 0.1600 & 0.0844 & 0.0627 \\ 0.0456 & 0.0342 & 0.0323 & 0.0235 & 0.0246 \end{bmatrix} \tag{31}$$

$$b_i = \begin{bmatrix} 0.25 & 0.5 & 1 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 \end{bmatrix} \tag{32}$$

1) ANALYSIS OF OPTIMIZATION RESULTS

In TABLE 2, the DWOA achieved significantly higher values for both the maximum and average convergence accuracies

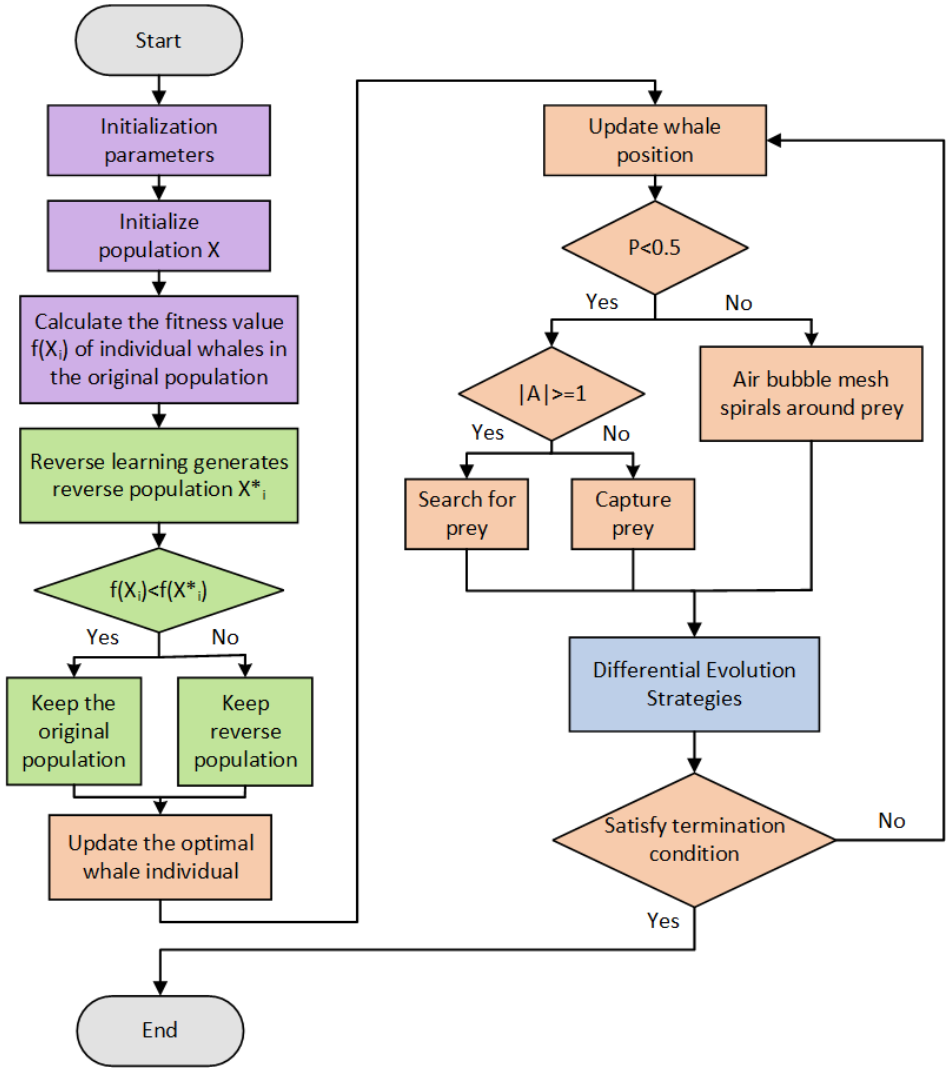

FIGURE 6. The algorithm flow of DWOA.

on unimodal test functions F1 and F2 when compared with the other five optimization algorithms with far smaller standard deviations. Although its performance on unimodal function F3 was comparable to that of GWO and PSO, the DWOA still achieved the theoretical optimal value. Its excellent performance on the unimodal functions indicated its excellent local development ability. The DWOA achieved the best values for multimodal functions F4 to F6, suggesting that it has an excellent global search ability and a strong capability of escaping local optimal solutions. Although the convergence results of the DWOA on fixed-dimension function F7 were equal to those of GWO, PSO, SSA, and WOA, its standard deviation was far smaller than that of the other algorithms, indicating good stability. On function F8, the DWOA achieved optimal optimization results. Overall, the enhanced algorithm can significantly improve its optimization capability.

2) OPTIMIZATION EFFICIENCY ANALYSIS

We used iteration data to generate comparative graphs for the iteration curves of the six algorithms and comprehensively analyze the DWOA performance, facilitating a more nuanced analysis of its optimization efficiency. FIGURE 9 reveals that the DWOA exhibits a superior convergence rate across all functions. While most optimization algorithms demonstrated comparable convergence rates on function F2, the DWOA excelled for its superior convergence accuracy relative to the other algorithms. It also depicted markedly superior convergence rates on other unimodal and multimodal functions. All algorithms exhibited relatively good convergence performances for fixed-dimension functions, which were easier to converge. However, FIGURE 9 and TABLE 2 demonstrate that the convergence performance of the DWOA continued to outperform the other algorithms. In other words, the DWOA is a highly effective optimization algorithm.

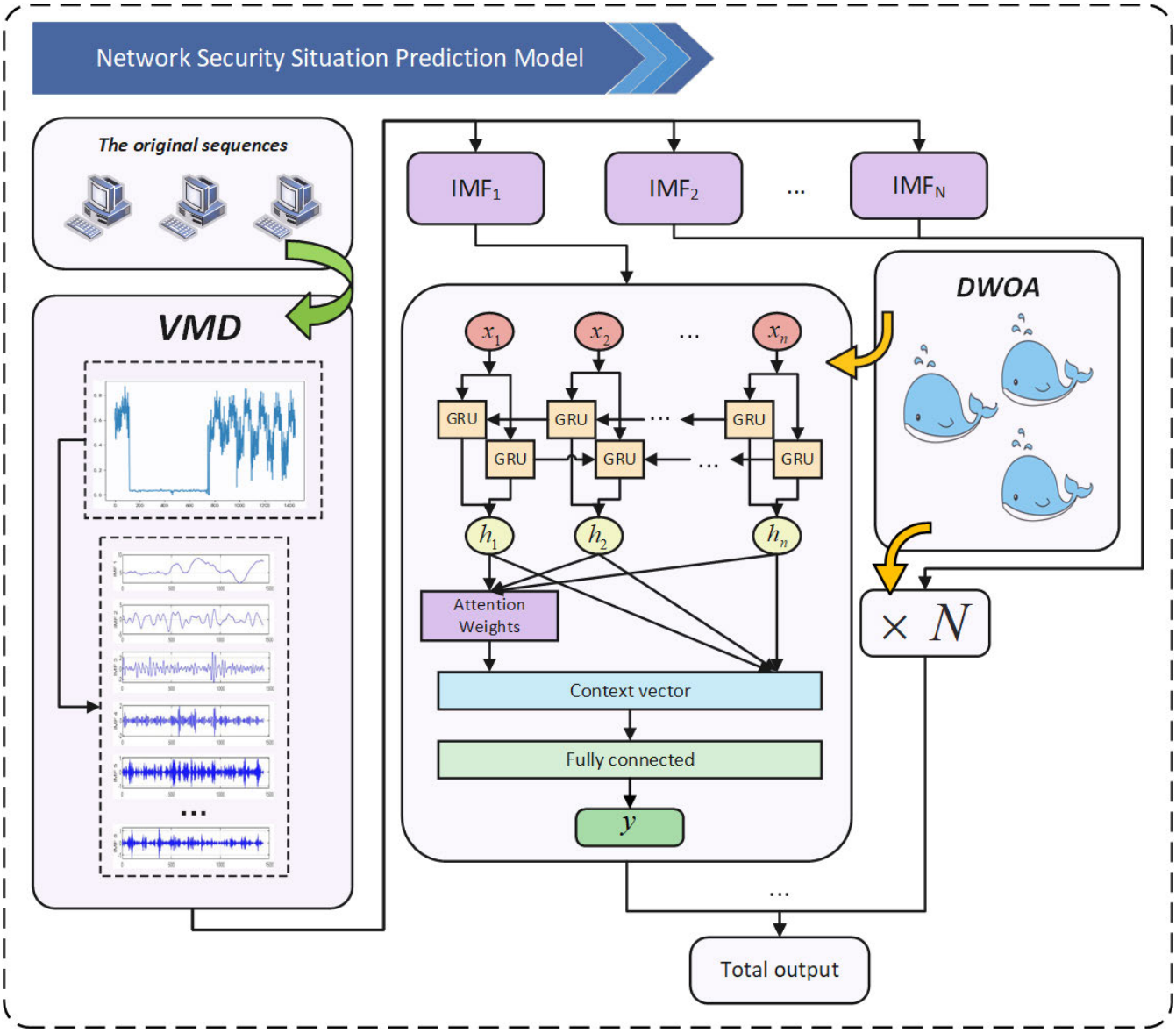

FIGURE 7. The overall structure of VMD-DWOA-BIGRU-ATTN.

B. SIMULATION ANALYSIS OF NETWORK SECURITY SITUATION PREDICTION

1) DATASET

The foundational dataset used herein was the UNSW-NB15 dataset [57] released by the University of New South Wales in Australia and the CIC-IDS2017 dataset [58] published by the Canadian Institute for Cybersecurity. The UNSW-NB15 dataset comprises both normal network traffic and various malicious traffic types and contains nine types of attacks and 49 network traffic features. TABLE 3 lists the specific attack types of the UNSW-NB15 dataset. TABLE 4 details the specific network traffic features of the UNSW-NB15 dataset. The CIC-IDS2017 dataset includes 8 types of attacks and 79 network traffic features. TABLE 5 presents the specific attack types of the CIC-IDS2017 dataset, while TABLE 6 enumerates the primary network traffic features of the CIC-IDS2017 dataset.

2) DATA PREPROCESSING

We divided the network security status into the five following levels to evaluate the cybersecurity situation in a more scientific manner: safety, low-risk, medium-risk, high-risk, and super-risk. TABLE 7 shows the specific information for each level. We referred herein to the Common Vulnerability Scoring System (CVSS) rating standard and a constructed index system to achieve our purpose. We propose the following indicators and their corresponding calculation formulas based on the existing quantification methods in this field:

(1) Threat level: This refers to the various risks and harms from external sources that a network system may face.

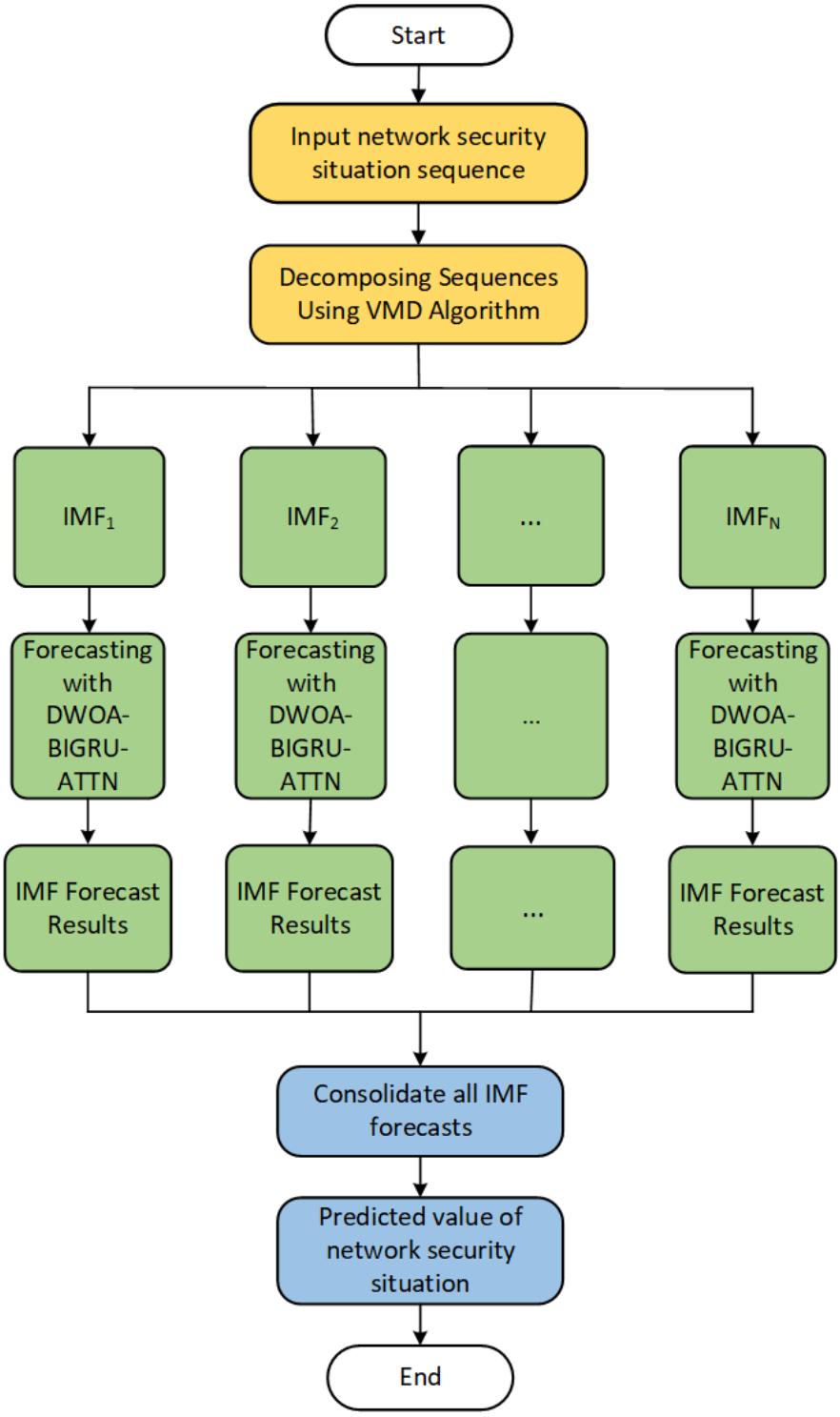

FIGURE 8. The process of model prediction.

In general, the most destructive cybersecurity issues are often triggered by external threats, which could lead to service disruptions or data loss. These threats primarily involve different types of network attacks and corresponding cybersecurity incidents. Therefore, in this study, we primarily focused on the severity of the selected attack types, alert frequency, and likelihood of security events occurring as secondary indicators under the primary indicator of the threat level. Taking the features of the UNSW-NB15 dataset as an example. The primary features involved in the metric are 'attack_cat' and

TABLE 1. Benchmark test functions.

Function formula	Dim	Range	F_{min}
$F_1(x) = \sum_{i=1}^{30} x_i^2$	30	[-100, 100]	0
$F_2(x) = \sum_{i=1}^{30} \lvert x_i \rvert + \prod_{i=1}^{30} \lvert x_i \rvert$	30	[-10, 10]	0
$F_3(x) = \sum_{i=1}^{30} (\sum_{j=1}^{i} x_j)^2$	30	[-100, 100]	0
$F_4(x) = -\sum_{i=1}^{30} (x_i \, sin(\sqrt{\lvert x_i \rvert}))$	30	[-500, 500]	-12569.5
$F_5(x) = \sum_{i=1}^{30} [x_i^2 - 10 \, cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]	0
$F_6(x) = -20 \, exp(-0.2 \sqrt{\frac{1}{30} \sum_{i=1}^{30} x_i^2}) - exp(\frac{1}{30} \sum_{i=1}^{30} cos \, 2\pi x_i) + 20 + e$	30	[-32, 32]	0
$F_7(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3} x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[-5, 5]	-1.0316285
$F_8(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5, 5]	0.0003075

'dstip' within the dataset. The total number of hosts has already been determined, as well as the importance level of each host. Furthermore, the threat level corresponding to each type of attack can be established based on existing literature [25]. The quantification formula is presented below.

$$T = \sum_{i=1}^{n} \sum_{j=1}^{k} \frac{10^{D_{ij}} Q_i C_{ij}}{M} \tag{33}$$

where T represents the threat score; n is the total number of hosts in the network system; k is the number of attack types; M is the number of attacks detected during a certain period; D_{ij} represents the harm level corresponding to the attack type received by host i for attack type j; Q_i is the importance level of host i; and C_{ij} is the number of times host i is attacked by attack j.

(2) Vulnerability: This mainly focuses on the security weaknesses existing within the network system, which are typically manifested as network topology vulnerabilities. These vulnerabilities often serve as entry points for network attacks. We can gain a deeper understanding of the network's security flaws by collecting information on vulnerabilities, including different types of known vulnerabilities, potential attack vectors, and possible risk levels. The importance of hosts also indicates the extent to which each network host may affect the overall system, that is, the potential level of impact on the entire network system if a particular host is attacked or exploited due to a vulnerability. We can more accurately assess the network vulnerability and identify which vulnerabilities and hosts pose a greater threat to the network security by synthesizing these two factors. In other words, the vulnerability information and the importance of hosts are the secondary indicators under the primary indicator of vulnerability that accurately describe the security vulnerabilities of the network system. Taking the features of the UNSW-NB15 dataset as an example. The primary feature involved in the metric within the dataset is 'Attack Reference', and the severity level of the vulnerabilities can be referenced from the CVSS vulnerability scoring system. The quantitative formula is as follows:

$$V = \sum_{i=1}^{n} \sum_{j=1}^{m} D_j Q_i S_{ij} \tag{34}$$

where V represents the vulnerability score; n is the total number of hosts in the network system; m is the number of vulnerability types; D_{ij} represents the harm level corresponding to the attack type received by host i for vulnerability j; Q_i is the importance level of host i; and S_{ij} is the score of vulnerability j on host i, which refers to the CVSS vulnerability rating.

(3) Reliability: In a network environment, the reliability is mainly influenced by the network system architecture and the potential issues with network devices. First, a well-designed network topology can reduce the likelihood of network interruptions, thereby enhancing the network stability. In addition, a robust network topology helps in load balancing and capacity planning, ensuring that network resources are efficiently used to further boost the network reliability. Second, the number of open ports on devices is another key factor affecting the network reliability. Open ports serve as communication channels between network devices and the external world

TABLE 2. The effectiveness of various optimization algorithms on benchmark testing functions.

Function	Algorithm	Best	Ave	Std
F1	GWO	9.16E-69	3.87E-66	7.94E-66
	GA	727.9122833	1568.429455	570.6731064
	PSO	3.47E-12	1.27E-08	3.10E-08
	SSA	6.91E-09	1.33E-08	3.15E-09
	WOA	5.33E-160	2.13E-126	1.15E-125
	DWOA	0	0	0
F2	GWO	5.34E-40	7.63E-39	7.87E-39
	GA	9.37395384	16.03737205	3.089641131
	PSO	1.09E-05	9.667193577	8.74960284
	SSA	0.002191973	0.746656462	0.789585262
	WOA	9.12E-96	3.55E-85	1.50E-84
	DWOA	4.42e-321	2.04E-282	0
F3	GWO	0	0	0
	GA	520	1425.466667	486.5596047
	PSO	0	0	0
	SSA	0	5	4.351245033
	WOA	0	0.033333333	0.179505494
	DWOA	0	0	0
F4	GWO	-7975.676491	-6440.76871	759.4845174
	GA	-10308.36227	-9277.012606	522.7009401
	PSO	-8481.192949	-6346.8188	1282.988412
	SSA	-9103.540636	-7515.648701	626.4211058
	WOA	-12569.48306	-11257.51005	1692.26586
	DWOA	-12569.4866	-12413.53084	450.2925474
F5	GWO	0	4.942978007	6.146151656
	GA	39.477742	56.15866498	10.71450894
	PSO	23.87905556	84.10160756	28.19761586
	SSA	18.90421705	56.14874323	16.73514021
	WOA	0	0	0
	DWOA	0	0	0
F6	GWO	1.47E-14	1.63E-14	2.86E-15
	GA	7.714790366	10.35385359	1.064163205
	PSO	1.75E-06	0.094429964	0.374559883
	SSA	2.69E-05	2.02232554	0.669587733
	WOA	4.44E-16	4.23E-15	2.42E-15
	DWOA	4.44E-16	1.63E-15	1.91E-15
F7	GWO	-1.031628453	-1.031628447	6.62E-09
	GA	-1.031628129	-1.0314461	0.00021472
	PSO	-1.031628453	-1.031628453	1.08E-09
	SSA	-1.031628453	-1.031628453	8.36E-15
	WOA	-1.031628453	-1.031628453	5.80E-11
	DWOA	-1.031628453	-1.031628453	6.60E-16
F8	GWO	0.000307487	0.003758389	0.007433032
	GA	0.000664761	0.001935785	0.001041467
	PSO	0.000341061	0.004720478	0.007243861
	SSA	0.000441634	0.001524349	0.003507074
	WOA	0.000307525	0.000628355	0.000290073
	DWOA	0.000307512	0.000596184	0.000333091

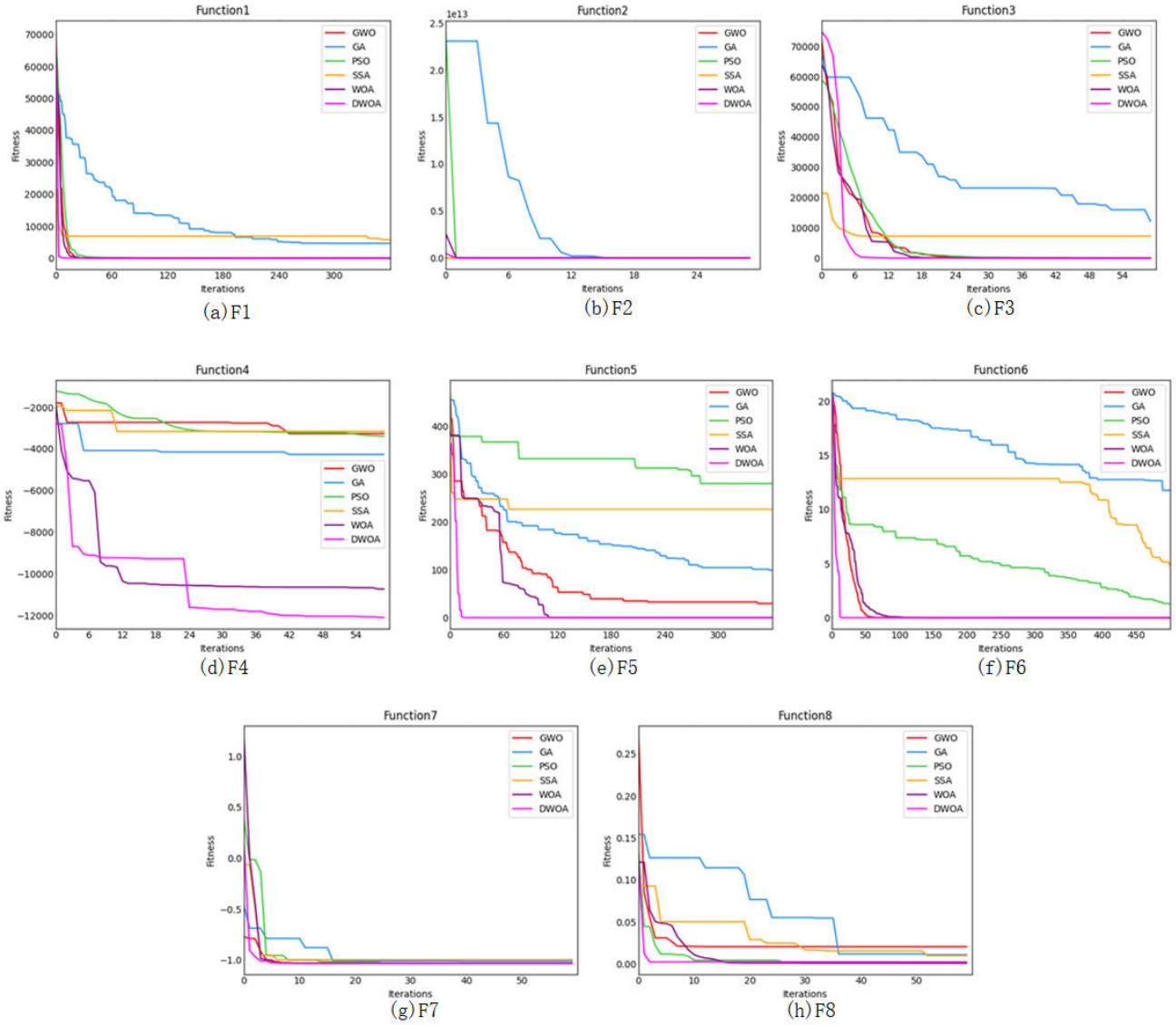

FIGURE 9. Convergence curves of six optimization algorithms on benchmark functions F1-F8.

and are potential points of security vulnerability. The network faces increased risks, including potential malicious attacks or unauthorized access, if there are too many open ports that are not properly secured and configured. This leads to data leaks, service outages, or system failures, which diminish the network reliability. In summary, optimizing the network topology and properly managing open device ports are critical for enhancing the network reliability. We mainly evaluated the network reliability by examining the network topology scores and the number of open device ports. Taking the features of the UNSW-NB15 dataset as an example. The primary features involved in the metric within the dataset are 'dsport' and 'dstip'. Additionally, the network topology structure has already been determined in the dataset. The calculation formula of R is as follows:

$$R = W_{TP} \times TP + W_{PR} \times PR \tag{35}$$

where W_{TP} and W_{PR} are the weights of the evaluation factors obtained through the AHP.

Network topology score: This indicator reflects the stability of the network topology structure. The higher the score, the poorer the network system stability. The quantitative formula is as follows:

$$tp_i = \begin{cases} 1.0, & 0 \le br < 3 \\ 0.5, & 3 \le br \le 5 \\ 0.1, & br > 5 \end{cases} \tag{36}$$

$$TP = \sum_{i=1}^{n} tp_i \tag{37}$$

where TP is the network topology score; br represents the number of branch nodes corresponding to each node in the network system; tp_i is the topology score corresponding to the ith node; and n represents the number of nodes. A larger number of branch nodes is generally more vulnerable to

TABLE 3. Types of attacks and their introduction (UNSW-NB15).

Attack type	Number of attacks	Description
Normal	2218761	Normal network traffic data.
Fuzzers	24246	Send randomly generated data to the network to make it suspend.
Analysis	2677	This attack includes port scanning, spam and other attack methods
Backdoors	2329	Bypass security mechanisms to gain unauthorized access to computers and data.
DoS	16353	By consuming network bandwidth and system resources by various means, the normal service of the normal system is paralyzed, thereby realizing the denial of normal user access to the service.
Exploits	44525	Attackers exploit security weaknesses to attack.
Generic	215481	A technique is effective against all block ciphers (with a specified block and key size) regardless of the block cipher's underlying structure.
Reconnaissance	13987	The attacker collects vulnerability information on the target system.
Shellcode	1511	Code executed to exploit software vulnerabilities.
Worms	174	A worm is a malicious program that can exploit system vulnerabilities to propagate itself through the network.

attacks, and a lower topology score indicates greater attack susceptibility.

Number of open ports: This metric reflects the number of open ports on all network devices, with a higher score indicating a greater number of open ports and a lower network reliability. The quantitative formula is as follows:

$$PR = \sum_{i=1}^{n} pr_i \tag{38}$$

where PR represents the score for the number of open ports; pr_i represents the number of open ports for the ith host; and n represents the total number of hosts within the network.

(4) Availability: Availability pertains to the capacity of the network system to function normally and provide essential services during a specific period. We primarily used the traffic fluctuation rate, data packet distribution pattern, and security event frequency as the indicators of network availability. First, the traffic fluctuation rate describes the speed at which the network data traffic changes. A low and stable traffic change rate usually indicates a relatively light network load, which helps in fully using the network resources and enhances the network availability. Conversely, highly fluctuating or sharply increasing traffic could trigger network congestion, increase latency, and degrade the service quality, thereby weakening the network availability. Second, packet distribution variations can imply whether or not the network is under attack network because attacks are often executed through data packets. A large amount of network traffic from abnormal sources observed in a short period likely signifies a network attack. Lastly, the security event frequency reveals the security risks the network system faces. These events may include service outages, data leaks, or malicious attacks, any of which would affect the network availability. Overall, the traffic change rate, packet distribution, and security event frequency collectively reflect the current level of the network system availability. Taking the features of the UNSW-NB15 dataset as an example. The primary features involved in the metric within the dataset are 'sbytes', 'dbytes', 'dstip', 'proto', and 'attack_cat'.

The quantification formula is as follows:

$$U = W_{fr} \times FR + W_{dp} \times DP + W_{sr} \times SR \tag{39}$$

where W_{fr}, W_{dp}, and W_{sr} are the weights of the evaluation factors calculated by the AHP.

The quantification formula for the change flow rate is as follows:

$$FR = \frac{f_t}{f_{t-1}} \tag{40}$$

where f_t is the traffic size in the current t time period, and f_{t-1} is the traffic size in the previous t time period.

The quantification formula for the packet distribution is as follows:

$$DP = ip \times dtp \tag{41}$$

where DP is the data packet distribution; ip is the number of different IP addresses; and dtp is the number of types of different protocols.

The quantification formula for the security incident frequency is as follows:

$$SR = \frac{se}{e} \tag{42}$$

where SR is the security event frequency; se is the number of security events; and e is the number of all events.

After quantifying the values of various indicators, the weight values $W = \{W_T, W_V, W_R, W_U\}$ of each indicator are calculated using the AHP [59], [60], [61], [62] and the

TABLE 4. Dataset features and their introduction (UNSW-NB15).

No.	Name	Type	Description
1	srcip	nominal	Source IP address
2	sport	integer	Source port number
3	**dstip**	nominal	Destination IP address
4	**dsport**	integer	Destination port number
5	**proto**	nominal	Transaction protocol
6	state	nominal	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	dur	Float	Record total duration
8	**sbytes**	Integer	Source to destination transaction bytes
9	**dbytes**	Integer	Destination to source transaction bytes
10	sttl	Integer	Source to destination time to live value
11	dttl	Integer	Destination to source time to live value
12	sloss	Integer	Source packets retransmitted or dropped
13	dloss	Integer	Destination packets retransmitted or dropped
14	service	nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service
15	Sload	Float	Source bits per second
16	Dload	Float	Destination bits per second
17	Spkts	integer	Source to destination packet count
18	Dpkts	integer	Destination to source packet count
19	swin	integer	Source TCP window advertisement value
20	dwin	integer	Destination TCP window advertisement value
21	stcpb	integer	Source TCP base sequence number
22	dtcpb	integer	Destination TCP base sequence number
23	smeansz	integer	Mean of the flow packet size transmitted by the src
24	dmeansz	integer	Mean of the flow packet size transmitted by the dst
25	trans_depth	integer	Represents the pipelined depth into the connection of http request/response transaction
26	res_bdy_len	integer	Actual uncompressed content size of the data transferred from the server s http service.
27	Sjit	Float	Source jitter (mSec)
28	Djit	Float	Destination jitter (mSec)
29	Stime	Timestamp	record start time
30	Ltime	Timestamp	record last time
31	Sintpkt	Float	Source interpacket arrival time (mSec)
32	Dintpkt	Float	Destination interpacket arrival time (mSec)
33	tcprtt	Float	TCP connection setup round-trip time, the sum of synack and ackdat .
34	synack	Float	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
35	ackdat	Float	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	Binary	If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0
37	ct_state_ttl	Integer	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
38	ct_flw_http_mthd	Integer	No. of flows that has methods such as Get and Post in http service.
39	is_ftp_login	Binary	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	integer	No of flows that has a command in ftp session.
41	ct_srv_src	integer	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).
42	ct_srv_dst	integer	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43	ct_dst_ltm	integer	No. of connections of the same destination address (3) in 100 connections according to the last time (26).
44	ct_src_ ltm	integer	No. of connections of the same source address (1) in 100 connections according to the last time (26).
45	ct_src_dport_ltm	integer	No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).
46	ct_dst_sport_ltm	integer	No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
47	ct_dst_src_ltm	integer	No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).
48	**attack_cat**	nominal	The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
49	Label	binary	0 for normal and 1 for attack records

TABLE 5. Types of attacks and their introduction (CIC-IDS2017).

Attack type	Number of attacks	Description
BENIGN	2273097	Normal network traffic data.
DoS	252661	It can consumes the web server resources and take it down very fast.
DDoS	128027	Distributed Denial of Service
Port Scan	158930	Port scanning is a network attack technique used to detect open network ports on a target host.
Bot	1966	Bots are self-running Trojans that can execute external commands.
FTP-Patator	7938	Attack FTP (File Transfer Protocol) services using brute force methods.
SSH-Patator	5897	Using brute force method against SSH (Secure Shell Protocol) service.
Heartbleed	11	Heartbleed is a security vulnerability in the encryption library OpenSSL.
Infiltration	36	The process of using techniques to attack, crack, test, and securely evaluate the security controls of a network system

TABLE 6. Dataset features and their introduction (CIC-IDS2017).

No.	Name	Description
1	Dst Port	Target port number
2	Protocol	The protocol used by the packet
3	Timestamp	Packet timestamp
4	Flow Duration	Flow Duration
5	Tot Fwd Pkts	Total number of forward packets
6	Tot Bwd Pkts	Total number of backward packets
7	TotLen Fwd Pkts	Total length of all forward packets
8	TotLen Bwd Pkts	Total length of all backward packets
9	Fwd Pkt Len Max	Maximum forward packet length
10	Fwd Pkt Len Min	Minimum forward packet length
11	Fwd Pkt Len Mean	Mean forward packet length
12	Fwd Pkt Len Std	standard deviation forward packet length
13	Bwd Pkt Len Max	Maximum backward packet length
14	BWD Pkt Len Min	Minimum backward packet length
15	BWD Pkt Len Mean	Mean backward packet length
16	BWD Pkt Len Std	standard deviation backward packet length
17	Flow Byts/s	The byte rate of the stream
18	Flow Pkts/s	The packet rate of the stream
19	Flow IAT Mean	mean arrival time between flows
20	Flow IAT Std	Standard deviation of arrival times between flows
21	Flow IAT Max	Maximum arrival time between flows
22	Flow IAT Min	Minimum arrival time between flows
23	Fwd IAT Tot	Total value of arrival time between forward flows
24	Fwd IAT Mean	Mean arrival time between forward flows
25	Fwd IAT Std	Standard deviation of arrival times between forward flows
26	Fwd IAT Max	Maximum arrival time between forward flows
27	Fwd IAT Min	Minimum arrival time between forward flows
28	Fwd PSH Flags	TCP flag
29	Bwd PSH Flags	TCP flag
30	Fwd URG Flags	TCP flag
31	Bwd URG Flags	TCP flag
32	Fwd Header Len	forward head length
33	Bwd Header Len	backward head length
34	Fwd Pkts/s	Forward packet rate
35	Bwd Pkts/s	Backward packet rate
36	Pkt Len Min	Minimum packet length
37	Pkt Len Max	Maximum packet length
38	Pkt Len Mean	Average packet length
39	Pkt Len Std	Standard deviation of packet length
40	Label	The label of the flow, indicating whether it is normal traffic or some kind of attack

TABLE 7. Details regarding each level.

Situation level	Situation value range	Scalar implicature
Safety	[0, 0.2)	No abnormal network traffic was detected, and the network environment is stable without any security risks.
Low-risk	[0.2, 0.4)	The network experiences low-risk anomalous traffic and low-severity security vulnerabilities exist, but overall the network remains stable and operational.
Medium-risk	[0.4, 0.6)	As the number of abnormal traffic increases and medium-risk security vulnerabilities emerge, the normal operation of the network is beginning to be affected.
High-risk	[0.6, 0.8)	Numerous security incidents have occurred, with a marked increase in medium-risk incidents and the emergence of some high-risk vulnerabilities.
Super-risk	[0.8, 1.0)	The network environment is severely affected by a high volume of attack behavior and critical vulnerabilities.

importance of various factors in the calculation process of the Analytic Hierarchy Process was compared with reference to existing safety standard [63], [64]. The situational value is then calculated as follows:

$$SV = W_T \times T + W_V \times V + W_R \times R + W_U \times U \quad (43)$$

3) EVALUATING INDICATOR

We selected the four following indicators to evaluate the model performance:

Mean squared error (MSE): The MSE is the squared difference between the actual and predicted values calculated as follows:

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \quad (44)$$

Root mean squared error (RMSE): The RMSE is the square root of the ratio of the square of the deviation between the predicted and true values to the number of observations n calculated as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2}{N}} \quad (45)$$

Mean absolute percentage error (MAPE): The MAPE is the mean absolute percentage error between the predicted and observed values calculated as follows:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right| \quad (46)$$

R-squared (R^2) score: R^2 is a statistical measure used to determine the goodness of fit of a regression model. It represents the proportion of variance in the dependent variable that can be explained by the independent variables in the model.

$$R^2 = 1 - \frac{\sum_{i=1}^{m}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{m}\left(y_i - \bar{y}\right)^2} \quad (47)$$

R-squared (R^2) score: R^2 is a statistical measure for determining the goodness of fit of a regression model. It represents the proportion of variance in the dependent variable that can be explained by the independent variables in the model.

4) COMPARISON OF CLASSICAL MACHINE LEARNING ALGORITHMS

We validated the efficacy of the employed neural network model in forecasting network security situations by comparing it with four other machine learning models. As shown in the TABLE 8, BiGRU–ATTN achieved the best performance. The models' prediction accuracy slightly improved with the addition of the attention mechanism. On the UNSW-NB15 dataset, the predictive performance of BiGRU-ATTN showed significant enhancement compared to the BP neural network and SVR methods. The prediction performance of BiGRU–ATTN exhibited a significant enhancement compared to both the BP neural network and SVR methods. Compared to the LSTM, BiGRU–ATTN showed increases of 21.25%, 11.26%, 11.41%, and 9.60% in the MSE, RMSE, MAPE, and R^2, respectively. Compared to GRU, BiGRU–ATTN demonstrated increases of 21.53%, 11.42%, 10.25%, and 9.78% in the MSE, RMSE, MAPE, and R^2, respectively. Compared to the BP neural network, it displayed increases of 34.92%, 19.33%, 19.46%, and 21.08% in the MSE, RMSE, MAPE, and R^2, respectively. Finally, compared to SVR, BiGRU–ATTN exhibited increases of 38.66%, 21.68%, 21.81%, and 25.70% in the MSE, RMSE, MAPE, and R^2, respectively. In summary, BiGRU–ATTN outperformed the other models in predicting the network security situation. On the CIC-IDS2017 dataset, the BiGRU model showed improvements ranging from 14.84% to 37.32% in MSE, from 7.72% to 20.83% in RMSE, from 5.00% to 18.27% in MAPE, and from 4.60% to 17.68% in R^2 compared to other models. FIGURE 10 shows the results of the machine learning model comparison.

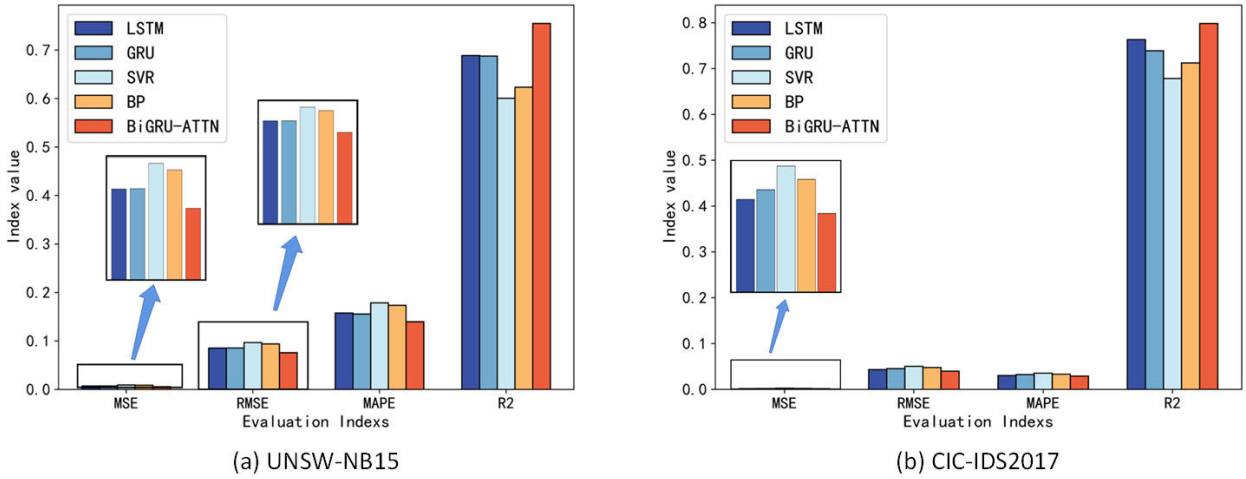

FIGURE 10. The comparison results of the machine learning model comparison.

TABLE 8. The effectiveness of various machine learning algorithms.

DATESET	MODEL	MSE	RMSE	MAPE	R^2
UNSW-NB15	LSTM	0.00729426	0.08540646	0.15766159	0.68888514
	GRU	0.00732028	0.08555863	0.15562083	0.68777555
	SVR	0.00936357	0.09676552	0.17863058	0.60062528
	BP	0.00882606	0.09394711	0.17342775	0.62355102
	BIGRU+ATTN	**0.00574388**	**0.07578838**	**0.13967069**	**0.75501213**
CIC-IDS2017	LSTM	0.00187196	0.04326613	0.03048005	0.76342410
	GRU	0.00206465	0.04543846	0.03243857	0.73907133
	SVR	0.00254327	0.05043087	0.03542787	0.67858403
	BP	0.00227622	0.04770971	0.03330358	0.71233431
	BIGRU+ATTN	**0.00159421**	**0.03992761**	**0.02895587**	**0.79852503**

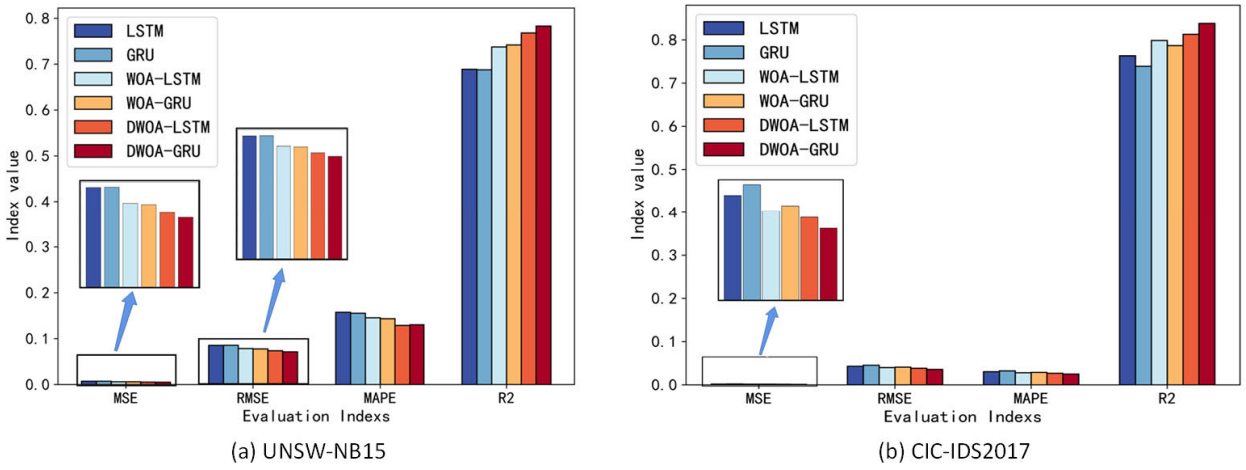

FIGURE 11. The comparison results of the effectiveness of the optimization algorithm.

5) THE EFFECTIVENESS OF OPTIMIZATION ALGORITHMS

The model performance was influenced by its parameters. An excellent parameter combination can be determined for an enhanced performance by employing an optimization algorithm to fine tune the hyperparameters. In this section, we aim to prove through experiments the effectiveness of the optimization algorithm in improving the performance. TABLE 9 shows the test results on the UNSW-NB15 dataset, indicating that compared to the unoptimized LSTM, the LSTM optimized by WOA improved the values of MSE, RMSE, MAPE, and R^2 by 15.68%, 8.17%, 7.63%, and 7.08%, respectively. Similarly, the WOA-optimized GRU

TABLE 9. The effectiveness of various optimization algorithms.

DATASET	MODEL	MSE	RMSE	MAPE	R^2
UNSW-NB15	LSTM	0.00729426	0.08540646	0.15766159	0.68888514
	GRU	0.00732028	0.08555863	0.15562083	0.68777555
	WOA+LSTM	0.00615021	0.07842329	0.14563901	0.73768122
	WOA+GRU	0.00605100	0.07778820	0.14359817	0.74191261
	DWOA+LSTM	**0.00543774**	**0.07374101**	**0.12894718**	**0.76806974**
	DWOA+GRU	**0.00508541**	**0.07131203**	**0.13027166**	**0.78309734**
CIC-IDS2017	LSTM	0.00187196	0.04326613	0.03048005	0.76342410
	GRU	0.00206465	0.04543846	0.03243857	0.73907133
	WOA+LSTM	0.00159354	0.03991921	0.02825287	0.79860980
	WOA+GRU	0.00168654	0.04106746	0.02902660	0.78685741
	DWOA+LSTM	**0.00147974**	**0.03846740**	**0.02688146**	**0.81299203**
	DWOA+GRU	**0.00127973**	**0.03577338**	**0.02498221**	**0.83826856**

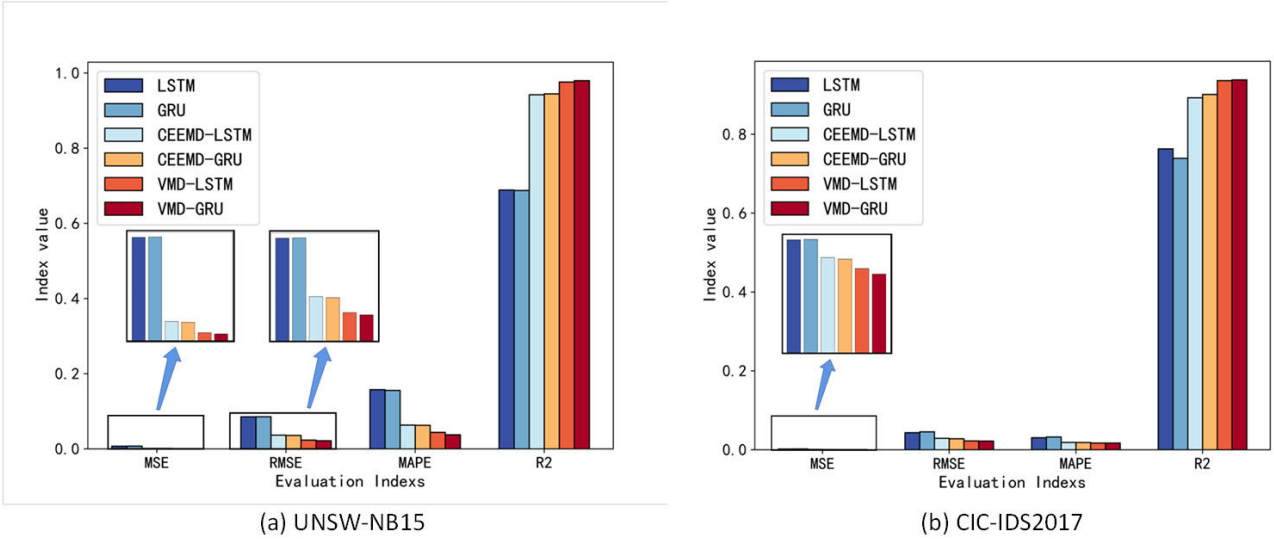

FIGURE 12. The comparison results of the effectiveness of the decomposition algorithm.

TABLE 10. The effectiveness of various decomposition algorithms.

DATASET	MODEL	MSE	RMSE	MAPE	R^2
UNSW-NB15	LSTM	0.00729426	0.08540646	0.15766159	0.68888514
	GRU	0.00732028	0.08555863	0.15562083	0.68777555
	CEEMD+LSTM	0.00135008	0.03674339	0.06336400	0.94241654
	CEEMD+GRU	0.00129225	0.03594788	0.06290328	0.94488297
	VMD+LSTM	**0.00055337**	**0.02352381**	**0.04409250**	**0.97639768**
	VMD+GRU	**0.00046922**	**0.02166155**	**0.03758239**	**0.97998671**
CIC-IDS2017	LSTM	0.00187196	0.04326613	0.03048005	0.76342410
	GRU	0.00206465	0.04543846	0.03243857	0.73907133
	CEEMD+LSTM	0.00084960	0.02914785	0.01861052	0.89262888
	CEEMD+GRU	0.00078224	0.02796855	0.01837298	0.90114139
	VMD+LSTM	**0.00050473**	**0.02246612**	**0.01717617**	**0.93621325**
	VMD+GRU	**0.00048911**	**0.02211581**	**0.01706560**	**0.93818694**

increased the MSE, RMSE, MAPE, and R^2 values by 17.34%, 9.08%, 7.73%, and 7.87%, respectively, compared to the unoptimized GRU. The DWOA-optimized LSTM also increased the MSE, RMSE, MAPE, and R^2 values by 11.92%, 6.15%, 11.10%, and 4.26%, respectively, compared to the WOA-optimized LSTM. Finally, the DWOA-optimized GRU increased the MSE, RMSE, MAPE, and R^2 values by 15.96%, 8.33%, 9.28%, and 5.55%, respectively, compared to the WOA-optimized GRU. TABLE 9 also shows the test results on the CIC-IDS2017 dataset, indicating that the model optimized by the optimization algorithm improved the values of MSE, RMSE, MAPE, and R^2 by an average

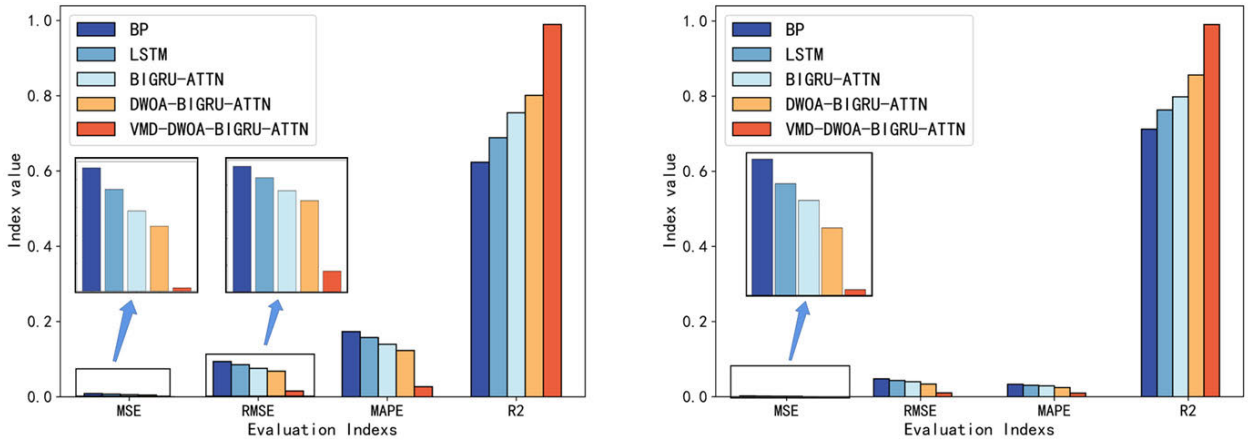

FIGURE 13. The comparison results of the effectiveness of various network security situation prediction methods.

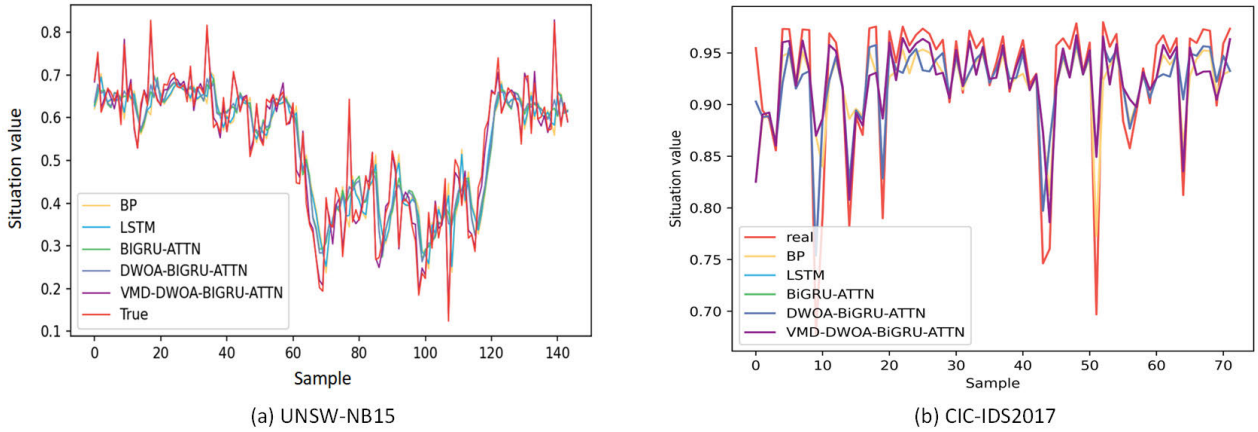

FIGURE 14. Comparison of situation prediction of different prediction models.

of 16.59%, 8.68%, 8.92%, and 5.54%, respectively. The enhanced optimization algorithm outperforms the traditional optimization algorithms, showing average improvements of 15.63%, 8.27%, 9.39%, and 4.17% in MSE, RMSE, MAPE, and R^2 values compared to the results of the conventional optimization algorithms. FIGURE 11 presents the comparison results of the effectiveness of the optimization algorithms.

The experimental results showed that the optimization algorithms can effectively improve the model prediction accuracy. The improved DWOA demonstrated a slight improvement over the original WOA in network model optimization.

6) THE EFFECTIVENESS OF DECOMPOSITION ALGORITHMS

We validated the efficacy of the VMD algorithm in enhancing the trend prediction accuracy by conducting experiments comparing the VMD-based ensemble model, complete ensemble EMD with adaptive noise (CEEMD)-based ensemble model, and standard model. TABLE 10 illustrates that the model's prediction accuracy was greatly improved after the decomposition algorithm was applied. TABLE 10 shows the test results on the UNSW-NB15 dataset, indicating that the CEEMD-LSTM combined model outperforms the standalone LSTM model by 81.49%, 56.98%, 59.81%, and 36.80% across the four performance metrics, respectively. The CEEMD–GRU combination model performed better than the single GRU model, yielding increases of 82.35%, 57.98%, 59.58%, and 37.38% in the MSE, RMSE, MAPE, and R^2 values, respectively. The VMD–LSTM combination model outperformed the single LSTM model with increases of 92.41%, 72.46%, 72.03%, and 41.74% in the MSE, RMSE, MAPE, and R^2 values, respectively. The VMD–GRU combination model outperformed the single GRU model with increases of 93.59%, 74.68%, 75.85%, and 42.49% in the MSE, RMSE, MAPE, and R^2 values, respectively. Meanwhile, the VMD–LSTM combination model outperformed the CEEMD–LSTM combination model, showing increases of 59.01%, 35.98%, 30.41%, and 3.61% in the MSE, RMSE, MAPE, and R^2 values, respectively. The VMD–GRU combination model outperformed the

TABLE 11. The effectiveness of various network security situation prediction methods.

DATASET	MODEL	MSE	RMSE	MAPE	R^2
UNSW-NB15	BP	0.00882606	0.09394711	0.17342775	0.62355102
	LSTM	0.00729426	0.08540646	0.15766159	0.68888514
	BIGRU+ATTN	0.00574388	0.07578838	0.13967069	0.75501213
	DWOA+BIGRU+ATTN	0.00465862	0.06825405	0.12324784	0.80130077
	VMD+DWOA+BIGRU+ATTN	**0.00024160**	**0.01554345**	**0.02700491**	**0.98969533**
CIC-IDS2017	BP	0.00227622	0.04770971	0.03330358	0.71233431
	LSTM	0.00187196	0.04326613	0.03048005	0.76342410
	BIGRU+ATTN	0.00159421	0.03992761	0.02895587	0.79852503
	DWOA+BIGRU+ATTN	0.00113455	0.03368314	0.02440934	0.85661638
	VMD+DWOA+BIGRU+ATTN	**0.00010426**	**0.01021061**	**0.00968821**	**0.99090127**

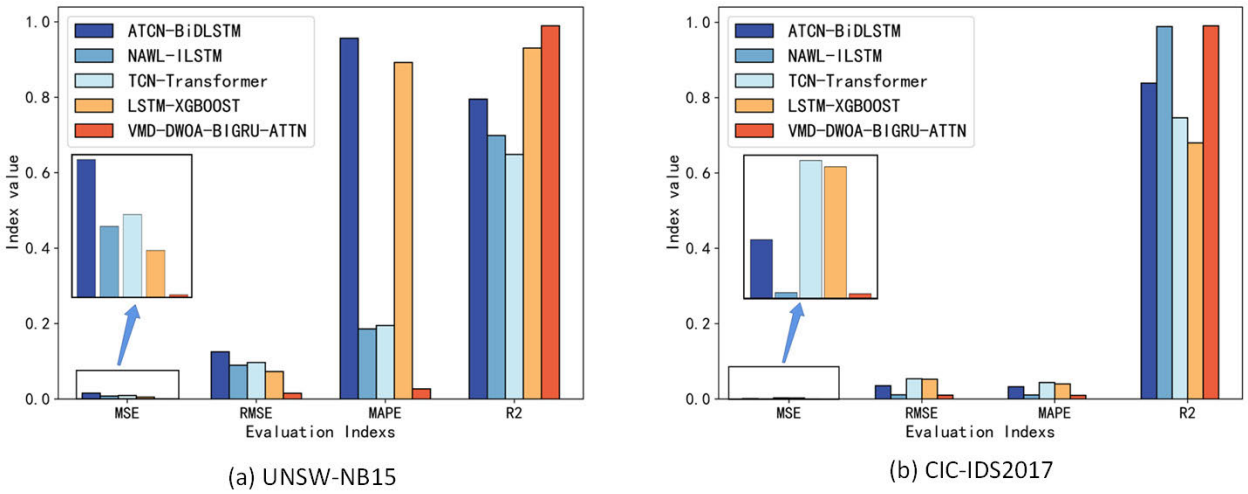

FIGURE 15. The comparison results of the effectiveness of different network security situation prediction baselines.

CEEMD–GRU combination model with increases of 63.69%, 39.74%, 40.25%, and 3.72% in the MSE, RMSE, MAPE, and R^2 values, respectively. TABLE 10 also shows the test results on the CIC-IDS2017 dataset. The model processed by CEEMD showed average improvements in the values of MSE, RMSE, MAPE, and R^2 by 58.36%, 35.54%, 41.15%, and 19.43%, respectively, compared to the original model. Furthermore, the model processed by VMD exhibited average enhancements of 39.03%, 21.93%, 7.42%, and 4.50% in MSE, RMSE, MAPE, and R^2 values respectively, compared to the model processed by CEEMD. FIGURE 12 shows the comparison results of the effectiveness of the decomposition algorithms. These results demonstrate the effectiveness of the VMD algorithm in improving the trend prediction accuracy.

Comparing the CEEMD and VMD decomposition algorithms, VMD exhibited a significantly greater improvement in optimization effectiveness. This finding validated the efficacy of the VMD algorithm used in this work in improving the trend prediction accuracy.

7) SIMULATION EXPERIMENT ANALYSIS OF VMD-DWOA-BIGRU-ATTN

TABLE 11 and FIGURE 13 lists the ablation test results on our proposed VMD-DWOA-BiGRU-ATTN. Among them, BiGRU-ATTN has previously been demonstrated to perform well in prediction tasks and is used as the base network for the proposed method. After integrating DWOA for optimization, the optimized network model shows average improvements of approximately 23.86%, 12.79%, 13.73%, and 6.70% in terms of MSE, RMSE, MAPE, and R^2 values respectively, compared to the base network model. However, as can be observed from FIGURE 14, the prediction accuracy of DWOA-BiGRU-ATTN is still not high in sections where the sequence has high non-stationarity, which is as expected. After incorporating the VMD method, compared to DWOA-BiGRU-ATTN, the VMD-DWOA-BiGRU-ATTN model saw average improvements of 92.81%, 73.46%, 69.20%, and 19.60% in MSE, RMSE, MAPE, and R^2 values respectively. It's evident that by introducing sequence processing techniques, there's a significant enhancement in prediction outcomes. As shown in FIGURE 14, our proposed method VMD-DWOA-BiGRU-ATTN not only maintains excellent prediction accuracy in stable time series sections but also retains good prediction accuracy in parts with high non-stationarity.

8) COMPARISON WITH EXISTING BASELINES

We further validated the effectiveness of VMD–DWOA–BiGRU–ATTN by comparing it with several other works

TABLE 12. Compare the predictive performance of different baselines.

DATASET	MODEL	MSE	RMSE	MAPE	R^2
UNSW-NB15	ATCN-BiDLSTM [34]	0.01569285	0.12527110	0.95679955	0.79533871
	NAWL-ILSTM [35]	0.00806502	0.08980545	0.18604729	0.69882509
	TCN-TRANSFORMER [25]	0.00941224	0.09701672	0.19492279	0.64851513
	LSTM-XGBOOST [32]	0.00531585	0.07290986	0.89243823	0.93067236
	VMD-DWOA-BIGRU-ATTN	**0.00024160**	**0.01554345**	**0.02700491**	**0.98969533**
CIC-IDS2017	ATCN-BiDLSTM	0.00124033	0.03521829	0.03261464	0.83886530
	NAWL-ILSTM	0.00012784	0.01130647	0.01078370	0.98884340
	TCN-TRANSFORMER	0.00290214	0.05387152	0.04379718	0.74672286
	LSTM-XGBOOST	0.00276811	0.05261280	0.04001976	0.68007516
	VMD-DWOA-BIGRU-ATTN	**0.00010426**	**0.01021061**	**0.00968821**	**0.99090127**

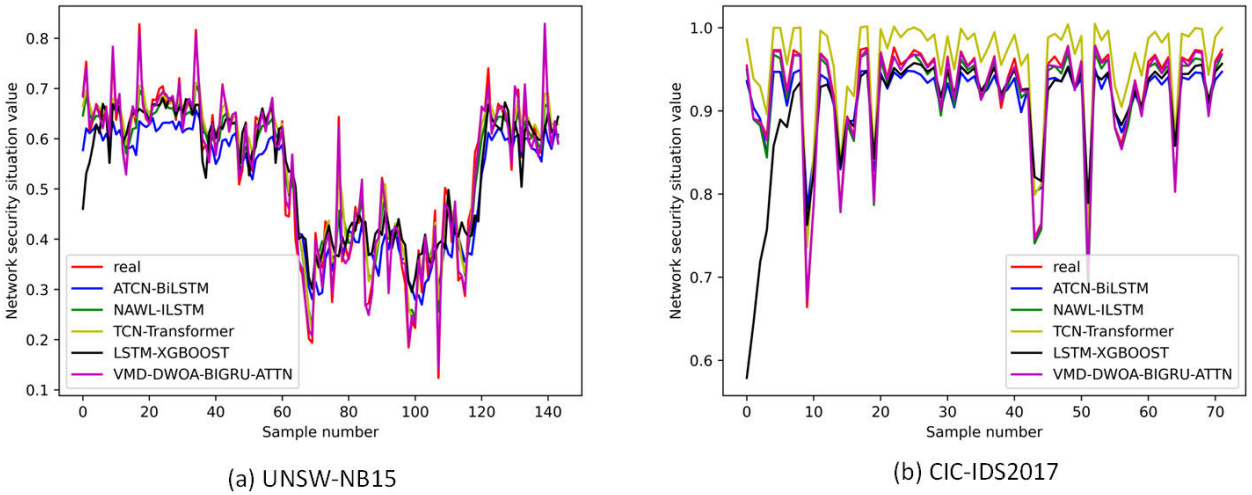

FIGURE 16. Comparison of situation prediction of different baselines.

related to the network security situation prediction. The baseline methods used for comparison are as follows:

LSTM–XGBoost [32]: An improved LSTM neural network model is established to predict the network security data, followed by the usage of the XGBoost model to assess the predicted data situation.

NAWL–ILSTM [33]: This utilizes an improved LSTM neural network model to establish a situational time series prediction model and the Look ahead method to enhance Nadam during the training process.

ATCN–BiDLSTM [34]: A network security situational prediction model is established, combining an attention mechanism-enhanced temporal convolutional network (ATCN) with a bidirectional long short-term memory (BiDLSTM) network.

TCN–Transformer [25]: It utilizes TCN combined with transformer as the network security situational data prediction model.

As shown in TABLE 12 and FIGURE 15. The test results on the UNSW-NB15 dataset indicate that compared to ATCN-BiDLSTM, the values of MSE, RMSE, MAPE, and R^2 improved by 98.46%, 87.59%, 97.18%, and 24.44%, respectively. When compared with those in ATCN–BiDLSTM, the MSE, RMSE, MAPE, and R^2 values improved by 98.46%, 87.59%, 97.18%, and 24.44%, respectively. Compared with those in NAWL–ILSTM, these values improved by 97.00%, 82.69%, 85.48%, and 41.62%, respectively. Compared with those in TCN–Transformer, the improvements were 97.43%, 83.98%, 86.15%, and 52.61%, respectively. When compared with those in LSTM–XGBoost, the improvements were 95.46%, 78.68%, 96.97%, and 6.34%, respectively. The test results on the CIC-IDS2017 dataset indicate that VMD-DWOA-BiGRU-ATTN outperforms other baseline models with improvements ranging from 18.44% to 96.41% in MSE, from 9.69% to 81.05% in RMSE, from 10.16% to 77.88% in MAPE, and from 0.21% to 45.70% in R^2. FIGURE 16 more intuitively displays the fit degree of the situation value curves of various baseline methods and the method proposed in this paper to the actual value curves, better illustrating the superior performance of our proposed method. In our experiments, the VMD–DWOA–BiGRU–ATTN model significantly outperformed the baseline models in terms of the situational prediction performance. This performance improvement was primarily be attributed to the synergistic operation of multiple components within the model. First, the model used VMD for data preprocessing. The VMD algorithm effectively reduced the data nonstationarity, which is crucial for

enhancing the model's predictive accuracy. Nonstationarity is a major challenge for traditional models in situational prediction; thus, preprocessing via VMD can significantly improve the model's predictive performance. Second, our model combined the DWOA with the BiGRU network. The DWOA can identify the optimal hyperparameter combinations, thereby enhancing its predictive capabilities. The BiGRU inclusion further strengthens the model's ability to process time series data, particularly when capturing long-term dependencies. Lastly, the model employs the ATTN to increase its sensitivity to key information. This attention mechanism can automatically identify and emphasize the time points or features more important for prediction, further enhancing the model's predictive ability. The experimental results demonstrate a significant performance advantage for this model compared to the baseline models.

VI. CONCLUSION

Research on the network security situational prediction is an important area of study in the network security field. Improving the prediction accuracy of network security situational sequences is significantly important in formulating security strategies and maintaining a stable network system operation. Network security situational sequences often exhibit nonstationary characteristics, making accurate predictions challenging using a single model. Therefore, this study developed a hybrid network security situational prediction method, called VMD–DWOA–BiGRU–ATTN, to address the low-accuracy issue of several existing network security situational prediction methods. The implementation of our approach started by decomposing the network security sequences using VMD, which transformed the original nonstationary time series into a set of relatively stationary subsequences. These subsequences were then predicted using the BiGRU–ATTN model optimized by the DWOA. Finally, the predicted results of each subsequence were aggregated. We evaluated the optimization performance of the DWOA on eight benchmark test functions. The experimental results demonstrated the significant superiority of the DWOA optimization performance over traditional optimization algorithms. The proposed prediction model was tested to assess its performance on the publicly available network security dataset, UNSW-NB15. The experimental results indicated that the VMD–DWOA–BiGRU–ATTN model outperformed several existing prediction models, achieving high MSE, RMSE, MAPE, and R^2 values on UNSW-NB15 (i.e., 0.00024160, 0.01554345, 0.02700491, and 0.98969533, respectively). The model significantly improved the prediction accuracy.

In this method, VMD effectively addressed the high nonstationarity issue in the network security situation sequences. Employing VMD for data preprocessing led to an average of ∼42% improvement in the R^2 value of the prediction results. BiGRU can handle sequence data with long-term dependencies and simultaneously extract features from past and future data, enabling a better sequence data processing that is difficult to handle using traditional methods. The attention mechanism also allowed the model to focus more on important time steps. By calculating the attention weights for each time step, the model automatically learned which time step's information was more crucial for predicting the target, thereby emphasizing these important time steps in the prediction process. The combined neural network BiGRU–ATTN outperformed the compared prediction models with approximately 10%–25% improvement in the R^2 value of the prediction results. The prediction method used the DWOA to optimize the model's hyperparameters, reducing the likelihood of the BiGRU–ATTN model getting stuck in local optima and improving the model's prediction ability and generalization. In summary, incorporating the optimization algorithm led to approximately 9%–11% improvement in the R^2 value of the prediction results.

This study combined sequence-processing techniques with optimization algorithms and deep learning techniques to effectively improve the network security situation prediction accuracy. Herein, we presented a new approach for predicting the network security situation sequences, offering valuable technical references for the network security professionals engaged in the network security situation prediction.

We have established a hybrid network security situation prediction model that accurately predicts security situations. However, there are still areas for improvement in subsequent research, mainly in two aspects. First, future research must consider simplifying the indicator construction system. This would allow testing of the proposed model on a broader range of datasets and enable better validation of the model's effectiveness and generalizability. It is also necessary to facilitate the application of the proposed prediction method to various forecasting scenarios. Second, in this work, we mainly focused on numerical prediction. However, the influencing factors in the actual network environment are more complex and variable. Therefore, the next step is to consider expanding numerical prediction to probabilistic prediction, that is, to predict the probability of each state of the network security situation so that changes in future information and uncertainties can be taken into account.

REFERENCES

[1] A. Thakare, E. Lee, A. Kumar, V. B. Nikam, and Y.-G. Kim, "PARBAC: Priority-attribute-based RBAC model for Azure IoT cloud," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 2890–2900, Apr. 2020, doi: 10.1109/JIOT.2019.2963794.

[2] M. A. Ferrag, L. Maglaras, A. Ahmim, M. Derdour, and H. Janicke, "RDTIDS: Rules and decision tree-based intrusion detection system for Internet-of-Things networks," *Future Internet*, vol. 12, no. 3, p. 44, Mar. 2020, doi: 10.3390/fi12030044.

[3] B. Zhang, M. Jia, J. Xu, W. Zhao, and L. Deng, "Network security situation prediction model based on EMD and ELPSO optimized BiGRU neural network," *Comput. Intell. Neurosci.*, vol. 2022, pp. 1–17, Jun. 2022, doi: 10.1155/2022/6031129.

[4] U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat, "A critical cybersecurity analysis and future research directions for the Internet of Things: A comprehensive review," *Sensors*, vol. 23, no. 8, p. 4117, Apr. 2023, doi: 10.3390/s23084117.

[5] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *J. Comput. Syst. Sci.*, vol. 80, no. 5, pp. 973–993, Aug. 2014, doi: 10.1016/j.jcss.2014.02.005.

[6] F. Cremer, B. Sheehan, M. Fortmann, A. N. Kia, M. Mullins, F. Murphy, and S. Materne, "Cyber risk and cybersecurity: A systematic review of data availability," *Geneva Papers Risk Insurance-Issues Pract.*, vol. 47, no. 3, pp. 698–736, Jul. 2022, doi: 10.1057/s41288-022-00266-6.

[7] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments," *Energy Rep.*, vol. 7, pp. 8176–8186, Nov. 2021, doi: 10.1016/j.egyr.2021.08.126.

[8] J.-B. Lai, H.-Q. Wang, X.-W. Liu, Y. Liang, R.-J. Zheng, and G.-S. Zhao, "WNN-based network security situation quantitative prediction method and its optimization," *J. Comput. Sci. Technol.*, vol. 23, no. 2, pp. 222–230, Mar. 2008, doi: 10.1007/s11390-008-9124-0.

[9] H. Alavizadeh, J. Jang-Jaccard, S. Y. Enoch, H. Al-Sahaf, I. Welch, S. A. Camtepe, and D. D. Kim, "A survey on cyber situation-awareness systems: Framework, techniques, and insights," *ACM Comput. Surv.*, vol. 55, no. 5, pp. 1–37, May 2023.

[10] A. Presekal, A. Stefanov, V. S. Rajkumar, and P. Palensky, "Attack graph model for cyber-physical power systems using hybrid deep learning," *IEEE Trans. Smart Grid*, vol. 14, no. 5, pp. 4007–4020, Sep. 2023, doi: 10.1109/TSG.2023.3237011.

[11] G.-Y. Hu and P.-L. Qiao, "Cloud belief rule base model for network security situation prediction," *IEEE Commun. Lett.*, vol. 20, no. 5, pp. 914–917, May 2016, doi: 10.1109/LCOMM.2016.2524404.

[12] D. Zhao and J. Liu, "Study on network security situation awareness based on particle swarm optimization algorithm," *Comput. Ind. Eng.*, vol. 125, pp. 764–775, Nov. 2018, doi: 10.1016/j.cie.2018.01.006.

[13] Z. Dong, X. Su, L. Sun, and K. Xu, "Network security situation prediction method based on strengthened LSTM neural network," *J. Phys., Conf. Ser.*, vol. 1856, no. 1, Apr. 2021, Art. no. 012056, doi: 10.1088/1742-6596/1856/1/012056.

[14] B. Zhang, S. Wang, L. Deng, M. Jia, and J. Xu, "Ship motion attitude prediction model based on IWOA-TCN-attention," *Ocean Eng.*, vol. 272, Mar. 2023, Art. no. 113911, doi: 10.1016/j.oceaneng.2023.113911.

[15] L. Run, L. X. Min, and Z. X. Lu, "Research and comparison of ARIMA and grey prediction models for subway traffic forecasting," in *Proc. Int. Conf. Intell. Comput., Autom. Syst. (ICICAS)*, Dec. 2020, pp. 63–67, doi: 10.1109/ICICAS51530.2020.00020.

[16] G. Wang, "Comparative study on different neural networks for network security situation prediction," *Secur. Privacy*, vol. 4, no. 1, p. e138, Jan. 2021, doi: 10.1002/spy2.138.

[17] J. Hu, D. Ma, C. Liu, Z. Shi, H. Yan, and C. Hu, "Network security situation prediction based on MR-SVM," *IEEE Access*, vol. 7, pp. 130937–130945, 2019, doi: 10.1109/ACCESS.2019.2939490.

[18] A. Pantazaras, S. E. Lee, M. Santamouris, and J. Yang, "Predicting the CO_2 levels in buildings using deterministic and identified models," *Energy Buildings*, vol. 127, pp. 774–785, Sep. 2016, doi: 10.1016/j.enbuild.2016.06.029.

[19] I. E. Livieris, S. Stavroyiannis, L. Iliadis, and P. Pintelas, "Smoothing and stationarity enforcement framework for deep learning time-series forecasting," *Neural Comput. Appl.*, vol. 33, no. 20, pp. 14021–14035, Oct. 2021, doi: 10.1007/s00521-021-06043-1.

[20] D. Liu, D. Niu, H. Wang, and L. Fan, "Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm," *Renew. Energy*, vol. 62, pp. 592–597, Feb. 2014, doi: 10.1016/j.renene.2013.08.011.

[21] H. Liu, C. Chen, H.-Q. Tian, and Y.-F. Li, "A hybrid model for wind speed prediction using empirical mode decomposition and artificial neural networks," *Renew. Energy*, vol. 48, pp. 545–556, Dec. 2012, doi: 10.1016/j.renene.2012.06.012.

[22] Z. Yang and J. Wang, "A hybrid forecasting approach applied in wind speed forecasting based on a data processing strategy and an optimized artificial intelligence algorithm," *Energy*, vol. 160, pp. 87–100, Oct. 2018, doi: 10.1016/j.energy.2018.07.005.

[23] A. A. Abdoos, "A new intelligent method based on combination of VMD and ELM for short term wind power forecasting," *Neurocomputing*, vol. 203, pp. 111–120, Aug. 2016, doi: 10.1016/j.neucom.2016.03.054.

[24] S. M. Zhang, B. X. Li, and B. Y. Wang, "The application of an improved integration algorithm of support vector machine to the prediction of network security situation," *Appl. Mech. Mater.*, vols. 513–517, pp. 2285–2288, Feb. 2014, doi: 10.4028/www.scientific.net/amm.513-517.2285.

[25] K. Yin, Y. Yang, C. Yao, and J. Yang, "Long-term prediction of network security situation through the use of the transformer-based model," *IEEE Access*, vol. 10, pp. 56145–56157, 2022, doi: 10.1109/ACCESS.2022.3175516.

[26] M. Adya and F. Collopy, "How effective are neural networks at forecasting and prediction? A review and evaluation," *J. Forecasting*, vol. 17, nos. 5–6, pp. 481–495, Sep. 1998, doi: 10.1002/(SICI)1099-131X(1998090)17:5/6<481::AID-FOR709>3.0.CO;2-Q.

[27] L. Suo, T. Peng, S. Song, C. Zhang, Y. Wang, Y. Fu, and M. S. Nazir, "Wind speed prediction by a swarm intelligence based deep learning model via signal decomposition and parameter optimization using improved chimp optimization algorithm," *Energy*, vol. 276, Aug. 2023, Art. no. 127526, doi: 10.1016/j.energy.2023.127526.

[28] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[29] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J. Big Data*, vol. 2, no. 1, p. 1, Feb. 2015, doi: 10.1186/s40537-014-0007-7.

[30] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[31] H. Zhang, C. Kang, and Y. Xiao, "Research on network security situation awareness based on the LSTM-DT model," *Sensors*, vol. 21, no. 14, p. 4788, Jul. 2021, doi: 10.3390/s21144788.

[32] L. Shang, W. Zhao, J. Zhang, Q. Fu, Q. Zhao, and Y. Yang, "Network security situation prediction based on long short-term memory network," in *Proc. 20th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2019, pp. 1–4, doi: 10.23919/APNOMS.2019.8893096.

[33] C. S. Z. Jiang, "Network security situation prediction method based on NAWL-ILSTM," *Comput. Sci.*, vol. 46, no. 10, pp. 161–166, Oct. 2019, doi: 10.11896/jsjkx.180901820.

[34] C. Yao, Y. Yang, J. Yang, and K. Yin, "A network security situation prediction method through the use of improved TCN and BiDLSTM," *Math. Problems Eng.*, vol. 2022, pp. 1–15, Oct. 2022, doi: 10.1155/2022/7513717.

[35] G. Zhao, L. Huang, L. Li, and Y. Zhang, "Prediction of industrial network security situation based on noise reduction using EMD," *Mobile Inf. Syst.*, vol. 2022, pp. 1–14, Mar. 2022, doi: 10.1155/2022/2594000.

[36] H. Xiaoyan, L. Bingjie, S. Jing, L. Hua, and L. Guojing, "A novel forecasting method for short-term load based on TCN-GRU model," in *Proc. IEEE Int. Conf. Energy Internet (ICEI)*, Sep. 2021, pp. 79–83, doi: 10.1109/ICEI52466.2021.00020.

[37] Q. Zhu, F. Zhang, S. Liu, Y. Wu, and L. Wang, "A hybrid VMD–BiGRU model for rubber futures time series forecasting," *Appl. Soft Comput.*, vol. 84, Nov. 2019, Art. no. 105739, doi: 10.1016/j.asoc.2019.105739.

[38] D. Zhao, P. Shen, and S. Zeng, "ALSNAP: Attention-based long and short-period network security situation prediction," *Ad Hoc Netw.*, vol. 150, Nov. 2023, Art. no. 103279, doi: 10.1016/j.adhoc.2023.103279.

[39] X. Kong, X. Du, G. Xue, and Z. Xu, "Multi-step short-term solar radiation prediction based on empirical mode decomposition and gated recurrent unit optimized via an attention mechanism," *Energy*, vol. 282, Nov. 2023, Art. no. 128825, doi: 10.1016/j.energy.2023.128825.

[40] L. Liao, H. Li, W. Shang, and L. Ma, "An empirical study of the impact of hyperparameter tuning and model optimization on the performance properties of deep neural networks," *ACM Trans. Softw. Eng. Methodol.*, vol. 31, no. 3, pp. 1–40, Jul. 2022.

[41] F. Li, Z. Wan, T. Koch, G. Zan, M. Li, Z. Zheng, and B. Liang, "Improving the accuracy of multi-step prediction of building energy consumption based on EEMD-PSO-informer and long-time series," *Comput. Electr. Eng.*, vol. 110, Sep. 2023, Art. no. 108845, doi: 10.1016/j.compeleceng.2023.108845.

[42] M. A. A. Al-qaness, A. A. Ewees, and M. Abd Elaziz, "Modified whale optimization algorithm for solving unrelated parallel machine scheduling problems," *Soft Comput.*, vol. 25, no. 14, pp. 9545–9557, Jul. 2021, doi: 10.1007/s00500-021-05889-w.

[43] S. M. Bozorgi, M. R. Hajiabadi, A. A. R. Hosseinabadi, and A. K. Sangaiah, "Clustering based on whale optimization algorithm for IoT over wireless nodes," *Soft Comput.*, vol. 25, no. 7, pp. 5663–5682, Apr. 2021, doi: 10.1007/s00500-020-05563-7.

[44] M. M. Saafan and E. M. El-Gendy, "IWOSSA: An improved whale optimization salp swarm algorithm for solving optimization problems," *Expert Syst. Appl.*, vol. 176, Aug. 2021, Art. no. 114901, doi: 10.1016/j.eswa.2021.114901.

[45] W. Yankai, W. Shilong, L. Dong, S. Chunfeng, and Y. Bo, "An improved multi-objective whale optimization algorithm for the hybrid flow shop scheduling problem considering device dynamic reconfiguration processes," *Expert Syst. Appl.*, vol. 174, Jul. 2021, Art. no. 114793, doi: 10.1016/j.eswa.2021.114793.

[46] L. Jain, R. Katarya, and S. Sachdeva, "Opinion leader detection using whale optimization algorithm in online social network," *Expert Syst. Appl.*, vol. 142, Mar. 2020, Art. no. 113016, doi: 10.1016/j.eswa.2019.113016.

[47] J. Anitha, S. I. A. Pandian, and S. A. Agnes, "An efficient multilevel color image thresholding based on modified whale optimization algorithm," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 115003, doi: 10.1016/j.eswa.2021.115003.

[48] R. Kushwah, M. Kaushik, and K. Chugh, "A modified whale optimization algorithm to overcome delayed convergence in artificial neural networks," *Soft Comput.*, vol. 25, no. 15, pp. 10275–10286, Aug. 2021, doi: 10.1007/s00500-021-05983-z.

[49] I. Aljarah, H. Faris, and S. Mirjalili, "Optimizing connection weights in neural networks using the whale optimization algorithm," *Soft Comput.*, vol. 22, no. 1, pp. 1–15, Jan. 2018, doi: 10.1007/s00500-016-2442-1.

[50] H. Peng, W.-S. Wen, M.-L. Tseng, and L.-L. Li, "A cloud load forecasting model with nonlinear changes using whale optimization algorithm hybrid strategy," *Soft Comput.*, vol. 25, no. 15, pp. 10205–10220, Aug. 2021, doi: 10.1007/s00500-021-05961-5.

[51] S. Mian Qaisar, S. I. Khan, K. Srinivasan, and M. Krichen, "Arrhythmia classification using multirate processing metaheuristic optimization and variational mode decomposition," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 1, pp. 26–37, Jan. 2023, doi: 10.1016/j.jksuci.2022.05.009.

[52] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1597–1600, doi: 10.1109/MWSCAS.2017.8053243.

[53] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates, 2017. Accessed: Jul. 19, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[54] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.

[55] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modeling, Control Autom. Int. Conf. Intell. Agents, Web Technol. Internet Commerce*, Nov. 2005, pp. 695–701, doi: 10.1109/cimca.2005.1631345.

[56] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997, doi: 10.1023/a:1008202821328.

[57] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. IEEE Military Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6, doi: 10.1109/MilCIS.2015.7348942.

[58] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy*, 2018, pp. 108–116.

[59] Q. Dong-mei and F. Chun-shu, "Study on network security assessment based on analytical hierarchy process," in *Proc. Int. Conf. Electron., Commun. Control (ICECC)*, Sep. 2011, pp. 2320–2323, doi: 10.1109/ICECC.2011.6066475.

[60] Y. Shuping and G. Yingyan, "Network security situation quantitative evaluation based on the classification of attacks in attack-defense confrontation environment," in *Proc. Chin. Control Decis. Conf.*, Jun. 2009, pp. 6014–6019, doi: 10.1109/CCDC.2009.5195279.

[61] L. Xiao, Y. Qi, and Q. Li, "Information security risk assessment based on analytic hierarchy process and fuzzy comprehensive," in *Proc. Int. Conf. Risk Manage. Eng. Manage.*, Nov. 2008, pp. 404–409, doi: 10.1109/icrmem.2008.71.

[62] T. L. Saaty, "Decision making with the analytic hierarchy process," *Int. J. Services Sci.*, vol. 1, no. 1, p. 83, Jan. 2008, doi: 10.1504/ijssci.2008.017590.

[63] *GB/T 20984-2007*. Accessed: Oct. 1, 2023. [Online]. Available: https://openstd.samr.gov.cn/bzgk/gb/newGbInfo?hcno=D7E8428EEFB1EBA4ED9AA78601AD5434

[64] *GB/T 20984-2022*. Accessed: Oct. 1, 2023. [Online]. Available: https://openstd.samr.gov.cn/bzgk/gb/newGbInfo?hcno=FDA38AB7D08A715C6B6D69DFDEABB2C0

SHENGCAI ZHANG was born in Lanzhou, Gansu, China, in 1982. He received the master's degree in signal and information processing from the Lanzhou University of Technology, Lanzhou, in 2009, where he is currently pursuing the Ph.D. degree in control theory and control engineering. He is an Associate Professor with the School of Cyber Security, Gansu University of Political Science and Law. His research interests include information security, artificial intelligence, and intelligent optimization.

QIMING FU was born in Weifang, Shandong, China, in 1999. He received the B.S. degree in computer science and technology from the Qilu University of Technology (Shandong Academy of Sciences), Jinan, China, in 2021. He is currently pursuing the master's degree in cybersecurity with the Gansu University of Political Science and Law. His main research interests include network security and artificial intelligence.

DEZHI AN was born in Shaoxing, Zhejiang, China, in 1973. He is currently a Professor with the School of Cyber Security, Gansu University of Political Science and Law, China. His main research interests include network security, public opinion analysis, and data mining.