Construction and Analysis of Network Cloud Security Situation Awareness System Based on DBN-DE Algorithm

Yunfei Zhang*, Xingzhi Xu and Yayuan Shi

State Grid Wuxi Power Supply Company; Wuxi, Jiangsu, 214000, China
E-mail: yunfzz123@126.com
**Corresponding Author*

Received 02 December 2023; Accepted 29 December 2023;
Publication 09 April 2024

Abstract

With the wide application of cloud computing, network security has become the primary issue in cloud environment. This study focuses on constructing a system focusing on network cloud security situation awareness, and combines deep belief network with differential evolution algorithm to improve the perception and analysis capability of network cloud security. Differential evolution algorithm is used to optimize DBN weights and parameters to improve system performance and generalization ability. In the aspect of system performance evaluation, the effectiveness of the system is verified by a series of experiments. The experimental results show that the system based on DBN-DE has excellent performance in network attack detection and can quickly identify various potential threats. The system also has a low false alarm rate, which can reduce the frequency of administrator intervention, and improve the efficiency and reliability of network cloud security.

Keywords: Cloud computing, network security, deep belief network, differential evolution algorithm.

Journal of Cyber Security and Mobility, Vol. 13_3, 439–460.
doi: 10.13052/jcsm2245-1439.1335

1 Introduction

Cloud computing is a revolutionary technology that provides individuals and businesses with a flexible, efficient and cost-effective way to store, manage and access data and run applications. The rise of cloud computing is rooted in the construction of large-scale data centers that support various cloud services with powerful computing and storage resources [1]. These services include infrastructure-as-a-Service (IaaS), platform-as-a-Service (PaaS), and software-as-a-service (SaaS), which enable users to benefit from the flexibility to scale their computing resources based on demand. The success of cloud computing is that it provides highly scalable resources and immediate service delivery, enabling enterprises to respond more nimbly to market demands. However, cloud computing also brings new security challenges, as sensitive data and business-critical applications are moved to the cloud environment, and therefore data and applications are potentially compromised.

Cloud security issues include, but are not limited to, data breaches, identity theft, denial of service attacks, malware spread, and insider threats. These threats can result in data loss, service unavailability, reputational damage and legal liability. Therefore, it is very important to ensure the network security of cloud environment. Traditional network security methods and tools are often unable to meet the needs of the cloud environment, because the cloud computing environment is highly dynamic, and the rapid expansion and reduction of resources will make the traditional security strategy less flexible [2, 3]. This requires new security solutions that can sense threats, attacks, and vulnerabilities in the network in real time and take appropriate measures to counter these threats.

Deep belief Network (DBN) is a multi-level neural network with excellent feature extraction and pattern recognition capabilities [4, 5]. DBN is an unsupervised learning algorithm that can automatically learn abstract feature representations of input data. The DBN consists of multiple layers of neurons, each connected to the front and back layers of neurons. This deep connection structure allows DBNS to capture complex relationships and patterns in data. An important advantage of DBN is its ability to handle large-scale, high-dimensional data, which is particularly important in network cloud security situational awareness because network data often contains a large number of characteristics and information. DBN can extract key features from this data for threat detection and situation analysis.

Differential evolution algorithm (DE) is a global optimization algorithm based on population search, which searches for the optimal solution by simulating individual selection and variation in the natural evolution process [6, 7]. A notable feature of DE is that it introduces differential operations into the solution space to generate new individuals, which helps to better explore the solution space and find the global optimal solution. In networked cloud security situational awareness systems, DE can be used to optimize the weights and parameters of DBN. Through parameter optimization of DBN, the system can obtain better performance and generalization ability, so as to detect network threats more accurately [8, 9].

This study aims to develop a network cloud security situational awareness system, utilizing the integration of deep trust networks (DBN) and differential evolution (DE) algorithms. The main goal is to enhance awareness and analysis of network cloud security. DBN is a deep learning neural network that excels in feature extraction and pattern recognition, as well as extracting key information from a wide range of network data. As a supplement, the optimization algorithm DE fine tunes the weights and parameters of the neural network to improve system performance and generalization ability. The unique advantage of this comprehensive method lies in its effectiveness in processing large-scale data, which helps to more accurately detect network attacks and improve situational awareness capabilities.

This article delves into the complex details involved in system design and construction, and comprehensively explores the principles and advantages of the DBN-DE algorithm. Subsequently, a detailed discussion was conducted on the performance evaluation of the system, and a series of rigorous experiments were conducted to demonstrate its outstanding ability in network attack detection. The experimental results not only emphasize the advantages of the system, but also emphasize its ability to quickly identify various potential threats, coupled with a low false positive rate, thereby reducing unnecessary intervention by administrators. Importantly, this revision responds to the reviewer's suggestion to highlight the specific contributions and innovations of our system compared to existing research. The primary significance of this study is that it serves as a powerful tool to strengthen cloud computing environments from network attacks and protect data and applications. Looking ahead to the future, as cloud security faces new challenges, the insights and solutions provided by the system design in this article will play a crucial role in advancing the field of cloud security and contributing to ongoing network security research and practice.

2 Situational Awareness Model

2.1 Endsley Situational Awareness Model

Endsley believes that the process of situation awareness consists of situation detection, understanding and prediction [10, 11]. These three parts are not independent of each other, but are interrelated and progressive. The result of situation awareness can be used as the information support of situation understanding, and the result of situation understanding can be the basis of situation prediction. The whole process can be understood as the constant cognitive mapping of environmental state information in a certain space and time. In the first stage of the situation model, all the activities of the network system are extracted by correlation analysis of the original security data collected from the network. In the second stage, based on the characteristics of the activities identified in the first stage, the purpose of these activities and the possibility of successful realization of these activities are analyzed, and the overall security situation of the network is evaluated. The third stage is to predict and analyze the future security situation.

In 1995, Endsley further developed an application model with situation awareness as the core based on the situation model, as shown in Figure 1. During decision making, the situational awareness module constantly obtains the state set of the system network, evaluates the overall situation of the current network on the one hand, and predicts the future development trend on the other [12]. According to the situational awareness results and decision rules, decision makers make decisions to respond to the current network in order to eliminate the attack threat in the network [13, 14].

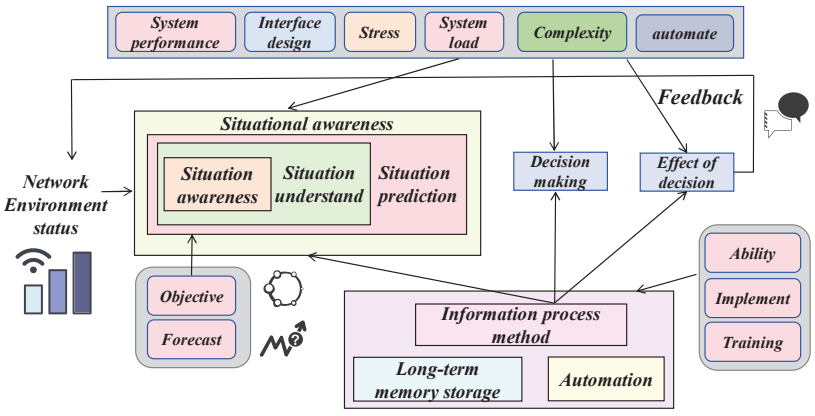

Figure 1 Endsely situational awareness application model.

2.2 Situation Awareness Model Based on OODA Decision Support

The OODA (Observe Orient Decision Act) loop designed by Boyd is a classic decision support model [15, 16]. OODA is a closed-loop process, including four stages: Observe, Orient, make decisions, Act. The observation traverses the physical domain and the information domain. It obtains raw state data from the physical domain to the information domain. Both orientation and decision are part of the cognitive domain, which constantly carries out cognitive mapping based on the information in the information domain and evaluates the situation to make decisions. Actions belong to the physical domain, and their execution decisions alter the existing state space and affect the subsequent observation phase. The whole process affects each other, and it goes on and on.

The OODA decision support model transforms the lower level of network security state into the highest level of inference. In order to realize network situational awareness, it is necessary to obtain network space state data in the observation stage. In the guidance stage, the overall analysis of the observed state data is emphasized to determine the impact on the network security state in the near future. In the decision-making stage, based on the observation of the evolving situation, the existing problems are classified and implicitly filtered to solve, and certain actions are selected to solve a specific problem; In the action phase, a decision can result in two types of actions. On the one hand, decision makers may choose to address the problem by implementing direct countermeasures in cyberspace, such as blocking the ports of firewalls, which will affect the state of the system and be visible through direct observation of network sensors; Decisions, on the other hand, may be countermeasures that lead to changes in the physical world, such as by regulating user behavior. At this stage, OODA is effectively closed by interacting with the observation-level physical environment. In the OODA decision support model, there is a dynamic cycle, so that the four stages have a clear division of labor and cooperate with each other, and each OODA cycle is constantly corrected and supplemented to make the situational awareness process more comprehensive.

2.3 Situational Awareness Model Based on JDL Data Fusion

Bass first applied situational awareness in the field of network security. In order to solve the problem that traditional intrusion detection systems cannot effectively detect professionally trained complex network attacks and the

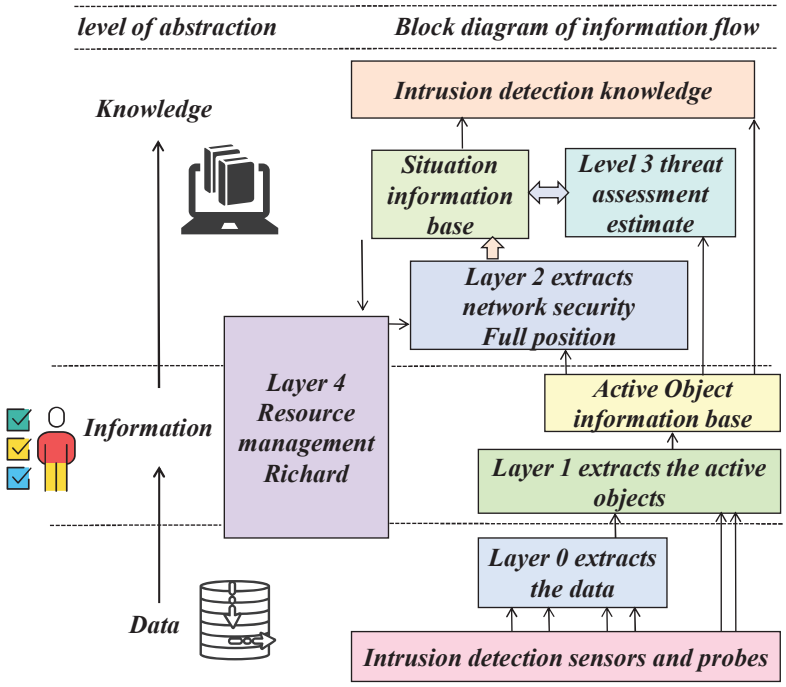

Figure 2 Endsely situational awareness application model.

high false positive rate of intrusion detection systems, he proposed multi-sensor fusion technology [17, 18]. On this basis, JDL data fusion model was used. The process of network security situation awareness is explained from the perspective of data fusion, which lays an important foundation for solving situation awareness by using data fusion technology. Figure 2 shows the network security situation awareness application model formed by Bass based on JDL data fusion model.

Continuing with the discussion on system security, at Level0, the initial phase involves the validation, filtering, and regularization of raw security data collected by network sniffers and intrusion detection systems. This data, comprising observation identifiers, times, and descriptions, undergoes further processing at Level1. Here, the observation data is formatted according to a generic framework, and individual observation objects are associated in both time and space. Additionally, weights are assigned based on their relevance, and objects are classified according to original intrusion primitives, ensuring accurate representation.

Moving to Level2, internal connections between objects are identified by analyzing object collections in terms of behavior, dependencies, common origin points, protocols, goals, attack rates, and other high-level properties. This step enhances situational knowledge by revealing the intricate relationships between various entities within the system. At Level3, leveraging existing situation knowledge and object information, the system is adept at identifying threat activities, understanding attack intentions, and conducting a comprehensive threat assessment of the entire network situation.

Level4 plays a crucial role in the entire process by orchestrating data fusion and formulating relevant strategies to optimize the effectiveness of data fusion. This level serves as a coordination and monitoring hub, ensuring seamless integration of information from multiple sources. It is instrumental in maintaining a holistic view of the network security landscape.

In terms of the system's ability to counteract attacks, the multilevel approach not only enhances situational awareness but also enables swift identification and response to potential threats. The interconnections of data at different levels facilitates a more nuanced understanding of attack patterns and potential vulnerabilities. Moreover, the system's adaptability to changing circumstances ensures that it remains resilient against evolving cyber threats [19, 20].

3 System Design and Construction

3.1 Principle of DBN-DE Algorithm

The DBN-DE algorithm combines two technologies, Deep Belief Network and Differential Evolution, to solve the optimization and feature extraction of complex problems [21, 22]. Here's how the DBN-DE algorithm works:

A Deep Belief Network (DBN) is a sophisticated deep learning model characterized by its multilayer structure composed of neurons. Neurons within each layer are fully connected, facilitating the transfer and extraction of information across different levels. DBNs consist of two primary types of layers: visible and hidden. The visible layer serves to represent the input data, while the hidden layer is dedicated to learning abstract features inherent in the data.

The construction of a DBN involves a layer-by-layer pre-training process, where each layer functions as a Restricted Boltzmann Machine (RBM). This approach enables the extraction of higher-order features from the data, contributing to the overall capability of the network. RBM, as the fundamental

component of DBN, is governed by an Energy Function expressed by Equation (1):

$$E(v,h) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i w_{ij} h_j \tag{1}$$

As shown in Formula (2), DBN generates the model through layer-by-layer Training, and then fine-tune the entire network through layer-by-layer Greedy Layer-Wise training.

$$P(v, h^{(1)}, h^{(2)}, \ldots, h^{(L)}) = \prod_{l=1}^{L} P(h^{(l)} v) P(v h^{(l+1)}) \tag{2}$$

Differential Evolution (DE) is a powerful global optimization algorithm designed for solving optimization problems within multidimensional parameter spaces. This algorithm mimics the principles of natural selection and evolution to search for the optimal solution. The key operations of DE involve mutation, crossover, and selection.

In the mutation phase, candidate solutions are chosen from the population, and differential variations are introduced in the parameter space to generate new potential solutions. This diversification strategy aims to explore the solution space comprehensively. Subsequently, in the crossover phase, the new solutions are combined with the current solutions to create offspring solutions. This recombination process helps to exploit the promising regions identified during the mutation phase.

The final selection operation determines which solutions will persist and form the next generation. DE operates on a population level, where each solution represents a potential optimal solution. Through iterative and evolutionary processes, DE endeavors to converge towards the global optimal solution. The steps of the DE algorithm can be summarized as follows:

(1) Initialize the population: N individuals are randomly generated as the initial population by Equation (3), and each individual is a D-dimensional vector.

$$X = \{x_1, x_2, \ldots, x_N\} \tag{3}$$

(2) Generate variants: According to Formula (4) for each individual, three different individuals a,b, and c are randomly selected.

$$v_i = x_a + F \cdot (x_b - x_c) \tag{4}$$

(3) For each dimension j, cross operations are performed with probability CR by Equation (5).

$$u_{ij} = \begin{cases} v_{ij} & if\ rand_j \leq CR\ or\ j = j_{rand} \\ x_{ij} & otherwise \end{cases} \tag{5}$$

(4) According to Formula (6), a new individual is selected by comparing the fitness of the variant and the target vector.

$$x_i^{new} = \begin{cases} u_i & if\ f(u_i) \leq f(x_i) \\ x_i & otherwise \end{cases} \tag{6}$$

The DBN-DE algorithm stands out for its capability to seamlessly integrate deep learning feature extraction and the global search of optimization algorithms, providing a powerful solution for addressing complex problems. This comprehensive approach opens up potential applications across diverse fields, including but not limited to image recognition, pattern recognition, and data classification. Through the synergistic utilization of DBN and DE, the algorithm exhibits excellent performance particularly in scenarios characterized by high dimensionality and nonlinearity.

Table 1 illustrates the step-by-step flow of the DBN-DE algorithm, and the construction steps are detailed as follows:

(1) DBN pre-training: Firstly, the layer-by-layer pre-training of DBN is used to initialize the weights and parameters of the network to better values. This helps DBN learn high-level feature representations from the data. Feature extraction for deep belief networks: Pre-trained DBNS are used to extract features from input data. These features can better

Table 1 DBN-DE algorithm flow

Procedure	Description
1	Initialize the structure and parameters of DBN, including the number of neurons in each layer, the learning rate, and the number of iterations
2	The unsupervised contrast divergence (CD) algorithm is used to train the DBN layer by layer, and the weight and bias of each layer are obtained
3	The DE algorithm is used to optimize the top-level weight and bias of DBN, and the optimal DBN model is obtained
4	The DBN model is used to approximate the objective function, and the approximate value of the function is obtained
5	The DE algorithm is used to find the optimal solution on the approximate function value, and the final optimization result is obtained

represent abstract information about the data and help with subsequent optimization and classification tasks.

(2) DE optimization: DE algorithms run on features extracted by deep belief networks to solve specific problems. By optimizing in the feature space, DE searches for the best parameter configuration to meet specific task requirements, such as classification or regression.

(3) Iteration: the DE algorithm improves the quality of the solution through iteration, and constantly searches the parameter space to find the best solution. Ultimately, it returns the parameter configuration with the best performance.

3.2 Data Preprocessing and Feature Extraction

When constructing a network cloud security situation awareness system based on DBN-DE algorithm, data preprocessing and feature extraction are crucial links [23, 24]. These steps are critical to preparing raw data for subsequent deep learning and optimization algorithms to process, as well as extracting key features for analysis and perception of cyber cloud security posture. Data acquisition and cleaning are the initial steps of data preprocessing. The system collects original data related to network cloud security, including network traffic logs, intrusion detection system (IDS) logs, system logs, and user behavior data, to ensure that the system has sufficient data for security situation awareness. The purpose of the data cleaning and de-noising stage is to deal with the noise, missing values and outliers that may exist in the original data. Data normalization and standardization is to overcome the problem of different data ranges and units among different data sources.

$$x_{norm} = \frac{x - min(x)}{max(x) - min(x)} \tag{7}$$

$$x_z = \frac{x - mean(x)}{std(x)} \tag{8}$$

By normalizing and standardizing Equations (7) and (8), the data is adjusted to have similar scales, which helps the deep learning model better understand the weights and features of the data.

$$covariance\_matrix = \frac{1}{n}\sum_{i=1}^{n}(data_i - mean(data))(data_i - mean(data))^T \tag{9}$$

Feature selection involves selecting the most informative features from the original data. Equation (9) is the process of calculating covariance matrix by principal component analysis (PCA). Feature selection can be done using statistical methods, correlation analysis, and the knowledge of domain experts to ensure that the selected features capture key information in network cloud security, such as traffic patterns, abnormal behavior, and attack characteristics. In the feature extraction stage, feature engineering can be carried out, that is, the construction of new features to enhance the expression of data, including aggregation, difference, discretization and other operations to create more informative features. These features help the system better understand and analyze network cloud security data. Finally, data encoding and transformation are performed to convert the raw data into a format suitable for the deep learning model, involving unique heat encoding, label encoding, normalization, etc., to ensure that the data can be efficiently input into the model.

3.3 Construction of Deep Belief Network

A Deep Belief Network (DBN) is a deep learning model used to learn feature representations in data. The DBN consists of multiple layers of neurons, including a visible layer, a hidden layer, and a final output layer. The process of building a DBN involves layer-by-layer pre-training and fine-tuning, the process of building a deep belief network:

(1) Initialization of visible layer and hidden layer:
The structure of DBN includes visible layer and hidden layer. The visible layer is typically used to represent the raw input data, while the hidden layer is used to learn abstract features of the data. These layers need to be initialized at the beginning of building the DBN.

(2) Pre-training layer by layer:
The construction of DBN usually adopts the way of layer by layer pre-training. Each hidden layer is a Restricted Boltzmann Machine (RBM). The layer-by-layer pre-training process consists of the following steps:

   a. Layer 1 RBM: The first RBM takes the raw input data (the data of the visible layer) and trains it to learn the weights and biases between the visible layer and the first hidden layer.
   b. After the completion of the first RBM, the output of its hidden layer is used as the input of the second RBM to continue the training. This process iterates until the desired hidden layer structure is built.

(3) Fine tuning:
Once the layer-by-layer pre-training is complete, fine-tuning is needed to further improve the DBN's performance. Fine-tuning can be done using backpropagation algorithms that update weights and parameters by backpropagating error signals.
The fine-tuning process can also include regularization tricks, such as Dropout, to avoid overfitting.

(4) Completion of training:
After the layer-by-layer pre-training and fine-tuning is completed, the DBN construction process is completed, and the model has the ability to learn the high-level feature representation in the data.

(5) Feature extraction and application:
Once the DBN is built, it can be used for feature extraction and various tasks such as classification, dimensionality reduction, generation, and more. The hidden layer of DBN can be used as a new representation of data, helping to solve complex problems.

3.4 System Architecture Based on Differential Evolution Algorithm

Differential Evolution plays a key role in the construction and analysis of network cloud security situation awareness system based on DBN-DE algorithm, providing powerful optimization and feature selection capabilities for the system [25, 26]. The application of DE algorithm is very important to improve the performance of network cloud security perception and analysis. Feature selection and dimension reduction, DE algorithm is widely used in feature selection to automatically determine which features in network cloud security data are most critical to the performance of the sensing system. Through DE algorithm, the system can identify and select the most informative features to achieve effective dimensionality reduction. This helps to reduce computational complexity and improve the training speed of the model, while preventing overfitting. For weight and parameter optimization, the DE algorithm can be used to optimize weights and parameters in deep belief networks [27, 28].

Expanding on the practical applications of the DBN-DE hybrid network architecture, Figure 3 illustrates its multi-layer neural network structure optimized by the DE algorithm. This optimization enhances the system's performance and generalization ability, making it well-suited for handling complex network cloud security data and improving overall accuracy.

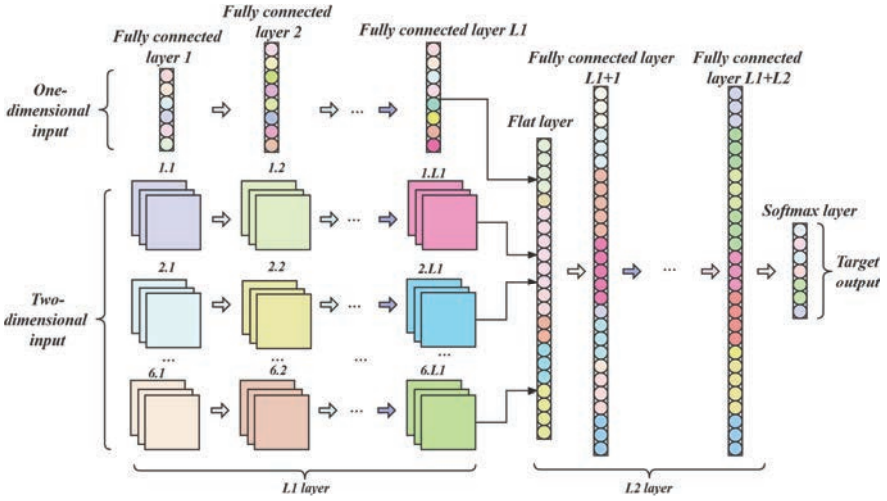

Figure 3 DBN-DE hybrid network architecture.

In the realm of intrusion detection and abnormal behavior analysis, the integration of DE algorithms proves instrumental. It enables the training and fine-tuning of intrusion detection systems, empowering them to adeptly identify new threats and attacks. The dynamic adjustment of intrusion detection rules and parameters facilitated by DE ensures the system's adaptability to evolving network security threats. This adaptability is critical for staying ahead of emerging risks and enhancing the system's responsiveness.

The application of DE extends to the optimization of hyperparameters in the model. Parameters such as learning rate, number of iterations, and batch size are fine-tuned using DE algorithms. This optimization ensures that the DBN-DE system can perform effectively across various scenarios, providing adaptability and robustness in different network environments.

The global search capability inherent in the DE algorithm is particularly noteworthy. This capability aids in finding optimal solutions without getting trapped in local minima, a crucial aspect when dealing with complex cyber cloud security issues. The system's ability to quickly and accurately identify a diverse range of potential threats is greatly enhanced by this global search capability, contributing to its efficacy in real-world scenarios.

In practical network environments, the DBN-DE hybrid network architecture finds application in scenarios such as continuous monitoring, threat detection, and response. Its adaptability and versatility make it well-suited

for dynamic and evolving network landscapes. As cyber threats continue to evolve, the system's capacity to dynamically adjust and optimize its parameters ensures its relevance and effectiveness in safeguarding cloud environments.

The DBN-DE hybrid network architecture, with its optimized structure and dynamic capabilities, proves to be a valuable asset in real-world applications. From intrusion detection to hyperparameter optimization, its multifaceted applications contribute to the enhancement of network cloud security situational awareness in diverse and dynamic environments.

4 Model Experiment and Result Analysis

4.1 Experimental Data Collection

This study collected comprehensive situational value data, which was calculated by integrating various dimensions, including basic operational indicators (CPU usage, memory usage, disk usage), vulnerability indicators (vulnerability CVSS score, quantity), threat indicators (alert threat level, quantity), and other related security indicators. Given the lack of existing datasets with similar characteristics, the comprehensive situational values were initially obtained by calculating the basic operational dimensions.

To verify the accuracy of the prediction model based on DBN-DE, this study established the experimental environment shown in Figure 4. The experimental setup includes collecting data from different dimensions of operations, vulnerabilities, and threats. The data collection process aims to capture the complexity of system behavior and potential security risks. Subsequently, the collected data is used to train and evaluate the DBN-DE model, ensuring its effectiveness in predicting and solving complex scenarios in the field of security.

In the experiment environment, IDSInformer attack software was used to simulate hacker attacks. Servers 1 to 5 were attacked. The data acquisition host deployed Snort intrusion detection system and Nessus vulnerability scanning system to collect network security data in the simulated environment. The collection time was 30 days, and a total of 4531 measured data were collected. Then, according to the evaluation method, the comprehensive situation data set is obtained. Among them, the situation set of the first 29 days is the training set, and the situation set of the 30th day is the test set. Table 2 shows the server vulnerability scanning table.

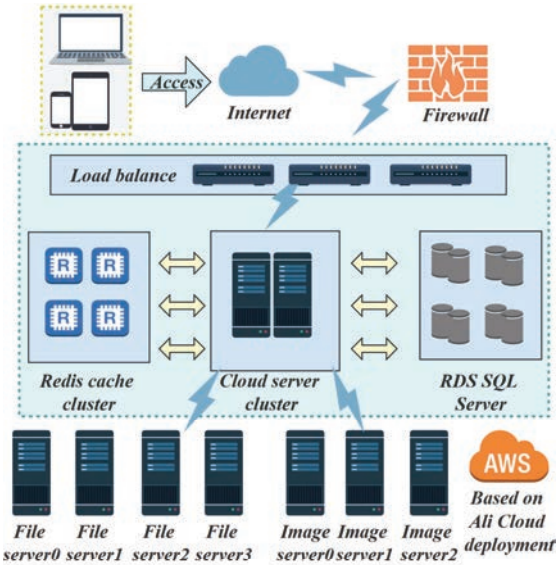

Figure 4 Cloud network deployment diagram.

Table 2 Server vulnerability scanning

Server Number	Running Service	CVE#	Vulnerability Description
1	Apache	CVE 2014-0098	Allows Remote Attackers To Cause a Denial of Service
2	Postgresql	CVE 2014-0063	Allow Remote Authenticated Users to Execute Arbitrary Code
3	Linux	CVE 2014-0038	Allow Local Users to Gain Priviliges Via a Recvmmsg System
		CVE 20131324	Allows Remote Attackers to Cause Stack-based Buffer Overflow
4	Bmc	CVE 2013-4782	Allows Remote Attackers to Bypass Authentication
5	Redius	CVE 20141878	Allows Remote Attackers to Cause a Segmentation Fault

4.2 Analysis of Prediction Results

In this paper, mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE) were used in prediction evaluation to evaluate the divine prediction model based on DBN-D E algorithm

[29, 30]. The formula is shown in Equations (10), (11) and (12).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_r - y_i'| \tag{10}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2} \tag{11}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|y_i - y_i'|}{y_i} \tag{12}$$

In the above equation, y_i is the true value, y_i' is the predicted value. Using the DBN-DE algorithm and 29 days' data as the training set, a prediction model was established to predict the network security situation within 24 hours of the last day, and the prediction result was obtained as shown in Figure 5. Comparing the curve of the real value and the predicted value on this day, it can be seen that the two curves almost coincide, which indicates that DBN-DE can accurately predict the network security situation.

In order to further study the advantages of DBN-DE algorithm, the prediction models of feedforward (BP) neural network and traditional LSTM neural network are established, and their prediction results are compared with those obtained by DBN-DE and the original time series situation data. The

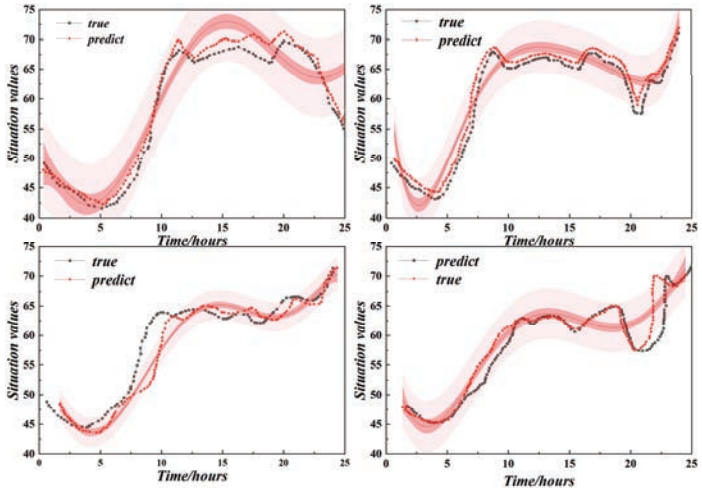

Figure 5 DBN-DE prediction results.

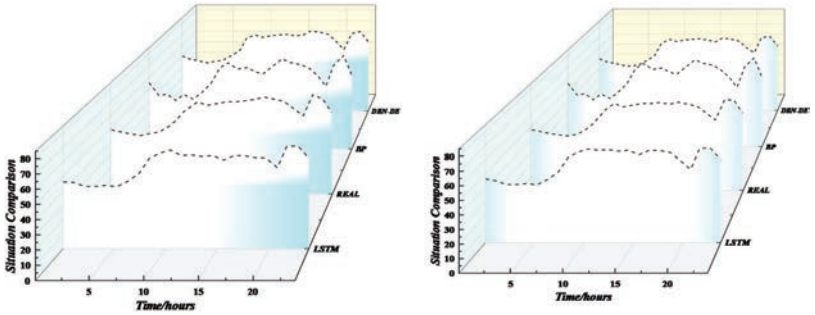

Figure 6 Comparison of situation values for different models.

Table 3 Error analysis of different models

Model	MAPE	RMSE	MAE
DBN-DE	1.887	1.378	1.144
LSTM	4.251	2.768	2.548
BP	4.841	3.564	2.859

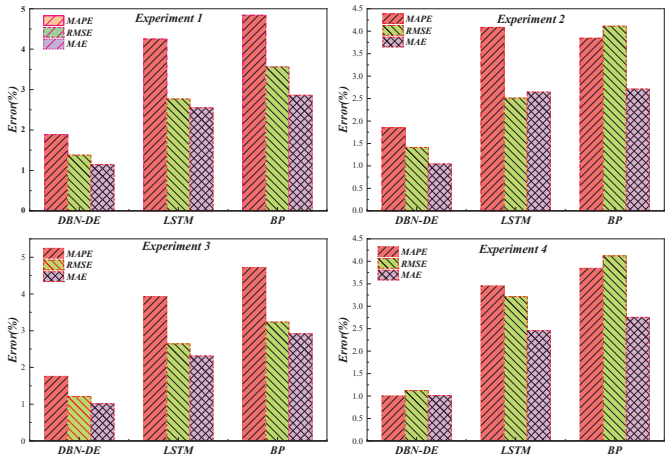

Figure 7 Error comparison of different models.

comparison results are shown in Figure 6. The prediction curve of DBN-DE model is closer to the real situation curve.

In order to evaluate the performance of the DBN-DE model more comprehensively, MAPE, RMSE and MAE of the three models in the prediction process were compared, as shown in Table 3. The error histogram shown in Figure 7 was further drawn, and it could be found more clearly that compared

with LSTM and BP neural network, MAPE, RMSE and MAE of the three models in the prediction process were compared. The MAPE, RMSE and MAE of DBN-DE model are much smaller than those of the other two methods, which further indicates the superiority of DBN-DE in the prediction of network security situation. Based on the above results, DBN-DE algorithm is selected as the situation prediction method of this system.

5 Conclusions

This paper realizes the perception and monitoring of the internal network security of a company based on the network cloud security situational awareness system. Relying on the creation process of the network security situational awareness system, the paper conducts a detailed investigation of the internal network topology and its business process. After in-depth research, the system design, requirement analysis and implementation process are carried out.

The main work of this paper includes the following parts:

(1) Analyze the functional and non-functional requirements of the system, introduce the key technologies of situation assessment and situation prediction, and introduce the intuitionistic fuzzy theory into the chromatography analysis method for the situation assessment of the system network, so as to make the assessment results more objective and accurate; By using the DBN-DE algorithm to predict the future trend, the experiment shows that the algorithm greatly improves the prediction accuracy.
(2) Analyze the whole system, model the system data and design the database through the class diagram and sequence diagram of each functional module of the system. On the basis of these, the network security situation awareness system is realized, which reduces the time and cost of the operation and maintenance personnel accessing the sensor, and improves the resource utilization and stability.
(3) In order to ensure that the system meets the requirements and availability, the data preparation module, network security situation awareness module and user management module of the system are tested and analyzed according to the functional design in the requirement analysis, and the non-functional requirements of the system are verified.

In future research, we will explore new ways to improve the robustness and scalability of the proposed system, and study predictive modeling

techniques combined with real-time threat intelligence to aid in the sustained development of the system.

References

[1] Bello S A, Oyedele L O, Akinade O O, et al. Cloud computing in construction industry: Use cases, benefits and challenges[J]. Automation in Construction, 2021, 122: 103441.

[2] Guo Q, Amin S, Hao Q, et al. Resilience assessment of safety system at subway construction sites applying analytic network process and extension cloud models[J]. Reliability Engineering & System Safety, 2020, 201: 106956.

[3] Jin Y, Chen W, Li H. A cloud-based approach to network security situational awareness[C]//International Conference on Signal Processing, Computer Networks, and Communications (SPCNC 2022). SPIE, 2023, 12626: 607–611.

[4] Xu H, Berres A, Yoginath S B, et al. Smart Mobility in the Cloud: Enabling Real-Time Situational Awareness and Cyber-Physical Control Through a Digital Twin for Traffic[J]. IEEE Transactions on Intelligent Transportation Systems, 2023, 24(3): 3145–3156.

[5] Torkura K A, Sukmana M I H, Cheng F, et al. Cloudstrike: Chaos engineering for security and resiliency in cloud infrastructure[J]. IEEE Access, 2020, 8: 123044–123060.

[6] Ignatious H A, El-Sayed H, Khan M A, et al. Analyzing Factors Influencing Situation Awareness in Autonomous Vehicles – A Survey[J]. Sensors, 2023, 23(8): 4075.

[7] Nassif A B, Talib M A, Nasir Q, et al. Machine learning for cloud security: a systematic review[J]. IEEE Access, 2021, 9: 20717–20735.

[8] Jiang D. The construction of smart city information system based on the Internet of Things and cloud computing[J]. Computer Communications, 2020, 150: 158–166.

[9] Aoudni Y, Donald C, Farouk A, et al. Cloud security based attack detection using transductive learning integrated with Hidden Markov Model[J]. Pattern Recognition Letters, 2022, 157: 16–26.

[10] Xu S, Ning J, Li Y, et al. Match in my way: Fine-grained bilateral access control for secure cloud-fog computing[J]. IEEE Transactions on Dependable and Secure Computing, 2020, 19(2): 1064–1077.

[11] Ageed Z S, Zeebaree S R M, Sadeeq M M, et al. Comprehensive survey of big data mining approaches in cloud systems[J]. Qubahan Academic Journal, 2021, 1(2): 29–38.

[12] Dinakarrao S M P, Jantsch A, Shafique M. Computer-aided arrhythmia diagnosis with bio-signal processing: A survey of trends and techniques[J]. ACM Computing Surveys (CSUR), 2019, 52(2): 1–37.
[13] Li W, Wu J, Cao J, et al. Blockchain-based trust management in cloud computing systems: a taxonomy, review and future directions[J]. Journal of Cloud Computing, 2021, 10(1): 1–34.
[14] Dwivedi R, Mehrotra D, Chandra S. Potential of Internet of Medical Things (IoMT) applications in building a smart healthcare system: A systematic review[J]. Journal of oral biology and craniofacial research, 2022, 12(2): 302–318.
[15] Oke A E, Kineber A F, Al-Bukhari I, et al. Exploring the benefits of cloud computing for sustainable construction in Nigeria[J]. Journal of Engineering, Design and Technology, 2023, 21(4): 973–990.
[16] Ali O, Shrestha A, Chatfield A, et al. Assessing information security risks in the cloud: A case study of Australian local government authorities[J]. Government Information Quarterly, 2020, 37(1): 101419.
[17] Chen B, Wu L, Wang H, et al. A blockchain-based searchable public-key encryption with forward and backward privacy for cloud-assisted vehicular social networks[J]. IEEE Transactions on Vehicular Technology, 2019, 69(6): 5813–5825.
[18] Zhang C. Design and application of fog computing and Internet of Things service platform for smart city[J]. Future Generation Computer Systems, 2020, 112: 630–640.
[19] Yaïci W, Krishnamurthy K, Entchev E, et al. Recent advances in Internet of Things (IoT) infrastructures for building energy systems: A review[J]. Sensors, 2021, 21(6): 2152.
[20] Butt U A, Mehmood M, Shah S B H, et al. A review of machine learning algorithms for cloud computing security[J]. Electronics, 2020, 9(9): 1379.
[21] Moudgil V, Hewage K, Hussain S A, et al. Integration of IoT in building energy infrastructure: A critical review on challenges and solutions[J]. Renewable and Sustainable Energy Reviews, 2023, 174: 113121.
[22] Vinoth S, Vemula H L, Haralayya B, et al. Application of cloud computing in banking and e-commerce and related security threats[J]. Materials Today: Proceedings, 2022, 51: 2172–2175.
[23] Elghaish F, Hosseini M R, Matarneh S, et al. Blockchain and the 'Internet of Things' for the construction industry: research trends and opportunities[J]. Automation in construction, 2021, 132: 103942.

[24] Bavle H, Sanchez-Lopez J L, Cimarelli C, et al. From slam to situational awareness: Challenges and survey[J]. Sensors, 2023, 23(10): 4849.
[25] Butt U A, Amin R, Aldabbas H, et al. Cloud-based email phishing attack using machine and deep learning algorithm[J]. Complex & Intelligent Systems, 2023, 9(3): 3043–3070.
[26] Wazid M, Das A K, Shetty S, et al. A tutorial and future research for building a blockchain-based secure communication scheme for internet of intelligent things[J]. IEEE Access, 2020, 8: 88700–88716.
[27] Wei P C, Wang D, Zhao Y, et al. Blockchain data-based cloud data integrity protection mechanism[J]. Future Generation Computer Systems, 2020, 102: 902–911.
[28] Medhane D V, Sangaiah A K, Hossain M S, et al. Blockchain-enabled distributed security framework for next-generation IoT: An edge cloud and software-defined network-integrated approach[J]. IEEE Internet of Things Journal, 2020, 7(7): 6143–6149.
[29] Shakya S. An efficient security framework for data migration in a cloud computing environment[J]. Journal of Artificial Intelligence, 2019, 1(01): 45–53.
[30] Kim H. 5G core network security issues and attack classification from network protocol perspective[J]. J. Internet Serv. Inf. Secur., 2020, 10(2): 1–15.

Biographies

Yunfei Zhang, Graduated from the Nanjing University of Posts and Telecommunications in 2003. Working in Ultra High Voltage Branch of State Grid Jiangsu Electric Power CO., LTD. His research interests include Electric Power auto-dispatching.

Xingzhi Xu, Graduated from the Adelaide University in 2005. Studied in Electrical Engineering. His research interests include Electric Power auto-dispatching, Artificial Neural.

Yayuan Shi, Graduated from the Southeast University in 2009. Studied in Electrical Engineering. Her research interests include Electric Power auto-dispatching, Integrated Relay.