Real-Time Information Security Situational Awareness in Big Data Networks Using an Improved C4.5 Decision Tree with Dynamic Feature Weighting and Hybrid Pruning

Lin Sun[1], Zhe Luo[2*]
[1]School of Information, Wuhan Vocational College of Software and Engineering, Wuhan 430205, China
[2]Jackie Chan Movie and Media College, Wuhan Institute of Design and Sciences, Wuhan 430205, China
E-mail: wids1818@163.com, 18971300780@163.com
[*]Corresponding author

Keywords: improved decision tree, situation awareness, feature weighting, mixed pruning, big data security

Received: June 12, 2025

With the rapid development of big data technology, network attacks are characterized by scale, concealment and intelligence. In this paper, an improved C4.5 decision tree algorithm (DW-C4.5) is proposed, and a real-time detection model is constructed by dynamic feature weighting (integrating random forest feature importance and information gain ratio optimization) and mixed pruning strategy (pre-pruning error rate threshold of 0.05+pruning cost complexity after pruning). Twelve kinds of attacks, such as DDoS, APT and zero-day exploitation, are tested on four public data sets (NSL-KDD, CIC-IDS2017 and UNSW-NB15) and one enterprise intranet log data set. The results show that the detection accuracy is 96.71%, which is 10.3 percentage points higher than that of traditional C4.5. The integrated Spark Streaming framework achieves a log throughput of 280,000 logs per second, and the false alarm rate is controlled below 3.12%. This method provides an efficient technical path for the dynamic security protection of massive network data.

Povzetek: Obravnavana je neučinkovitost klasičnega C4.5 pri varnostni analitiki velikih podatkov, predvsem pristranskost izbiranja značilk, prenasičenost dreves in slabo obvladovanje visokodimenzijskih omrežnih podatkov. Predlaga izboljšani DW-C4.5 z dinamičnim uteževanjem značilk (IGR+RFI+SHAP) ter hibridnim obrezovanjem. Model v realnem času doseže 96,71-odstotno točnost in nizko 3,12-odstotno lažno alarmiranje.

1 Introduction

With the acceleration of global digital transformation, cyberspace security has become the core component of national security strategy. According to IBM's Data Leakage Cost Report IBM 2023, the global average economic loss caused by a single data leakage incident is as high as 4.45 million US dollars, and the complexity of attacks has increased exponentially-advanced persistent threat (APT), zero-day exploitation, supply chain attacks and other new attack methods account for more than 67%. At the same time, the scale of network traffic data has entered the EB era, and the traditional rule-based security detection technology faces two core challenges: first, the static rule base is difficult to cope with the dynamic evolution of attack patterns, and the false alarm rate is generally higher than 15% [1]; Secondly, the real-time analysis ability of massive heterogeneous data (such as traffic logs, terminal behaviors, threat intelligence) is insufficient, resulting in an average threat response time (MTTR) of 6.2 hours [2]. In this context, the network information security situational awareness model, which integrates big data technology and machine learning algorithm, is becoming a key technical path to break the bottleneck of safe operation efficiency [3]. Traditional network security situational awareness technology has been difficult to meet the demand in real-time and accuracy. How to efficiently process massive data and accurately identify security threats has become a key issue to be solved urgently. The purpose of this study is to improve the efficiency and accuracy of information security situation awareness in big data networks by improving the decision tree algorithm, and to provide more powerful technical support for network security protection.

The Poseidon framework proposed by the team of Seyed K. Fayaz of Stanford University integrates reinforcement learning algorithm to realize the closed-loop control from threat detection to policy implementation, and shortens the response delay to 43 seconds in the simulation environment, but its assumption of relying on the perfect attack simulation environment limits the actual deployment effect [4-7]. However, there are still obvious defects in the mainstream international schemes: commercial products such as IBM QRadar rely too much on expert rule base, and it is difficult to identify zero-day attacks with hidden characteristics; However, the scheme based on deep learning is limited in its application in strict regulatory fields such as finance and government affairs because of its poor interpretability [8].

LightSA model proposed by Chen Chun, an academician of Zhejiang University, improves the

reasoning speed of the model to 150,000 TPS in Huawei cloud environment through feature layered compression technology, while maintaining the detection accuracy of 93.4%, but its detection sensitivity to time series attacks is insufficient [9]. The "Alibaba Cloud" system developed by Alibaba Cloud Security Team integrates the streaming computing engine with independent intellectual property rights, supports real-time analysis of daily average PB-level data, and has provided security services for more than 2,000 government units, but it lacks open academic theoretical support [10].

Comparative analysis of key related work is shown in Table 1.

Table 1: Comparative analysis of key related work

Scheme name	Core method	Test data set	Key indicators (accuracy/delay)	Main limitation
Poseidon framework	Reinforcement learning closed-loop control	Simulation environment data set	Response delay of 43 seconds	Depending on the perfect attack simulation environment, the actual deployment is limited.
IBM QRadar	Expert rule base	Business scenario data	Ambiguous/sub-second delay	It is difficult to identify zero-day attacks and rely on static rules.
LightSA model	Feature layered compression technology	Huawei cloud environment data	93.4%/ reasoning speed 150,000 TPS	The sensitivity of time series attack detection is insufficient
Alibaba Cloud system	Self-mainstream computing engine	PB-level daily average data	No explicit/real-time analysis support.	Lack of open academic theory support

Nevertheless, domestic research still faces two major pain points: First, the feature alignment error (18% on average) caused by protocol heterogeneity in the process of multi-source data fusion seriously restricts the detection efficiency [11]; Secondly, the existing decision tree algorithm has insufficient ability to express time series characteristics, and the recall rate is less than 70% when detecting complex attacks such as DNS hidden tunnels [12].

Focusing on the core challenges of information security situational awareness in big data networks, this study identified the following research questions:

RQ1: Can the dynamic feature weighting mechanism alleviate the traditional C4.5 preference for discrete features and improve the detection accuracy of complex attacks such as APT and zero-day exploitation?

RQ2: Can the hybrid pruning strategy reduce the complexity of the model and ensure the real-time detection performance (delay and throughput)?

RQ3: What is the generalization ability of the proposed model on multi-source heterogeneous data sets? Compared with the existing SOTA method, does it have advantages in accuracy, delay and false alarm rate?

2 Relevant theoretical basis

2.1 Network information security situation awareness

Cyber security situation awareness (CSSA) is a dynamic cognitive process, and its theoretical framework usually includes three stages: multi-source data fusion, threat situation assessment and risk prediction response. Based on the three-tier architecture of JDL (Joint Directors of Laboratories) data fusion model, this study extracts traffic behavior fingerprints (such as TCP flag bit distribution and DNS query entropy) through feature engineering [13]. The comprehensive risk index is calculated by combining threat intelligence, asset weight and vulnerability exposure surface, and the attack diffusion trend is evaluated by time series prediction (ARIMA) and propagation model (SEIR). Based on the situation evolution analysis of hidden Markov model (HMM), the probability of attacker's behavior transfer (for example, the transition probability from reconnaissance stage to penetration stage reaches 0.68) can be quantified, which provides a theoretical basis for active defense. The research further shows that the effectiveness of situational awareness model is strongly correlated with data quality (Pearson r=0.83), which highlights the key role of big data preprocessing process.

2.2 Decision tree algorithm principle

Decision tree algorithm realizes classification by recursively dividing feature space, and its core lies in the optimization of feature selection criteria and pruning strategy. In this study, the improved C4.5 algorithm is adopted, and its information Gain Ratio is calculated as formula (1):

$$\text{GainRatio}(D,A) = \frac{\text{InfoGain}(D,A)}{\text{SplitInfo}_A(D)} \quad (1)$$

This criterion effectively avoids the preference problem of ID3 algorithm for multi-valued features. Aiming at the high noise characteristics of network security data, a dynamic pruning strategy is proposed: when the number of node samples () is empirical coefficient), the split is terminated. Experiments show that this strategy can reduce the complexity of the model by 42% and only lose 1.3% accuracy. Compared with the integration methods such as random forest and GBDT, the single decision tree sacrifices part of the accuracy (F1-score decreases by about 5%), but its reasoning speed is increased by 7 times (single prediction takes less than 0.1ms), which is more suitable for real-time scene detection. $N < \alpha \cdot \sqrt{N_{\text{total}}}$ ($\alpha = 0.8$) experimental results show that this strategy can reduce the model complexity by 42% and only lose 1.3% accuracy. Compared with the integration methods such as random forest and GBDT, the single decision tree sacrifices part of the accuracy (F1-score decreases by about 5%), but its reasoning speed is increased by 7 times (single prediction takes less than 0.1ms), which is more suitable for real-time scene detection.

3 Improved decision tree algorithm design

3.1 traditional decision tree algorithm problem analysis

3.1.1 Feature selection deviation

The traditional C4.5 algorithm uses information Gain Ratio for feature selection, but its preference for discrete features significantly affects the classification effect of network security data. Taking network protocol type (discrete) and packet length (continuous) as examples, assuming that the feature set is, the calculation difference of information gain ratio can be quantified as: $\mathrm{F} = \{F_1, F_2, ..., F_d\}$, The calculation difference of information gain ratio can be quantified as follows(as shown in Formula 2):

$$\text{Bias}(F_i) = \frac{\log_2(V_i)}{\sqrt{N}} \quad (V_i: \text{Feature extraction number}, N: \text{Total sample number}) \quad (2)$$

Experiments show that (as shown in Table 2), in CIC-IDS2017 data set, the probability of false selection of discrete features by traditional algorithms reaches 39.2%.

Table 2: Feature selection deviation experiment (NSL-KDD Data Set)

Feature type	Candidate feature number	Selected times	Misselection rate
Discrete type	32	17.4	38.6%
successive type	51	6.8	13.2%

3.1.2 Insufficient pruning strategies

Traditional pruning methods (such as PEP) are prone to under-pruning in network security scenarios. The complexity of define that tree structure is (Formula 3):

$$\Omega(T) = \alpha \cdot |T| + \beta \cdot \sum_{t \in T} \text{depth}(t) \quad (3)$$

Where is the number of nodes and the depth of nodes. The average tree depth after traditional pruning is 9.7 layers, and the optimal depth should be controlled at 5~7 layers. $|T|$ is the number of nodes, $\text{depth}(t)$ is the node depth, $\alpha = 0.01$, $\beta = 0.05$ .Experiments show that the average tree depth after traditional pruning is 9.7 layers, and the optimal depth should be controlled at 5~7 layers.

3.1.3 high-dimensional data processing is inefficient

The time complexity of traditional algorithms increases exponentially under high-dimensional network data (); d>100)The time complexity increases exponentially (Formula 4):

$$T(d) = O(d \cdot n \log n) \quad (n: \text{sample number}) \quad (4)$$

Table 3 shows that when the feature dimension d=200, the training time reaches 7.3 times that of d=50.

Table 3: Relationship between time complexity and feature dimension

Dimension (d)	Actual time (multiple)	Theoretical prediction	Error (%)
50	1.0	1.5	+50%
100	2.4	3.0	+25%
200	7.3	10.0	+37%
250	12.6	15.0	+19%

3.2 improvement strategies and methods

3.2.1 Dynamic feature weighting mechanism (DFW)

As an integrated learning method, random forest improves the classification performance by constructing multiple decision trees and synthesizing their prediction results. In this process, the frequency at which each feature is selected when the tree node is split reflects its importance for classification decision. Based on this mechanism, RFI evaluates its contribution to the overall classification performance by calculating the average Gini impurity or error rate of features in random forests. This method makes it possible to examine the importance of features from a global perspective and capture the complex interaction effects between features.

Shapley additional explanations, as an explanatory tool, are used to reveal the marginal contribution of features to classification results. SHAP value is based on the concept of Shapley value in cooperative game theory, and its direct influence on the prediction results is quantified by calculating the average marginal contribution of each feature in all possible feature subsets. This method not only provides a quantitative evaluation of the importance of features, but also generates an intuitive explanatory chart to help users understand how features work together to influence classification decisions.

Characteristic weighted hyperparametric sensitivity analysis is shown in Table 4.

Calculation formula of feature weight (Formula 5):

Table 4: Characteristic weighted hyperparametric sensitivity analysis (CIC-IDS2017 Data Set)

λ1 (IGR)	λ2 (RFI)	λ3 (SHAP)	f1 score, f score, f measure	Accuracy rate
0.5	0.3	0.2	0.959	96.7%
0.6	0.2	0.2	0.948	95.9%
0.4	0.4	0.2	0.952	96.2%
0.5	0.2	0.3	0.955	96.5%

Note: λ1+λ2+λ3=1. Experiments show that the original combination (0.5,0.3,0.2) has the best performance.

Table 5: Balance between pruning depth and performance of sub-dataset

Data set	Depth after pruning (layer)	F1 score of test set	Model complexity (number of nodes)
NSL-KDD	six	0.942	1150
CIC-IDS2017	seven	0.959	1200
UNSW-NB15	five	0.938	1080
Enterprise intranet log	six	0.927	1120

Table 6: Relationship between cluster size and training performance

Number of cluster nodes	Training time (minutes)	AllReduce communication overhead (seconds)	speed-up ratio
eight	45.2	12.8	1.0×
16	24.5	10.3	1.8×
32	13.2	8.7	3.4×
64	7.1	6.2	6.4×

$$w_i = \frac{\lambda_1 \cdot \mathrm{IGR}_i + \lambda_2 \cdot \mathrm{RFI}_i + \lambda_3 \cdot \mathrm{SHAP}_i}{\sum_{j=1}^{d} (\lambda_1 \cdot \mathrm{IGR}_j + \lambda_2 \cdot \mathrm{RFI}_j + \lambda_3 \cdot \mathrm{SHAP}_j)} \quad (5)$$

Where, is the adjustment parameter. $\lambda_1 = 0.5$, $\lambda_2 = 0.3$, $\lambda_3 = 0.2$ to adjust the parameters.

3.2.2 Hybrid pruning strategy (HPS)

Balance between pruning depth and performance of sub-dataset is shown in Table 5.

A two-stage pruning method is proposed:

Stage 1: Pre-pruning

Set the dynamic error rate threshold (Formula 6):

$$\theta_{\mathrm{pre}} = \dot{o}_0 + \gamma \cdot \frac{\mathrm{depth}}{\max\_\mathrm{depth}} \quad (\dot{o}_0 = 0.05, \gamma = 0.1) \quad (6)$$

Stage 2: Post-pruning

Improved pruning formula of cost complexity (Formula 7):

$$\mathrm{CC}(T) = \frac{R(T) + \alpha \mid T \mid}{R_{\mathrm{emp}}(T)} + \beta \cdot \mathrm{Var}(T) \quad (7)$$

Where is the variance of node classification,,. $\mathrm{Var}(T)$ Classify variance for nodes, $\alpha = 0.01$, $\beta = 0.005$.

3.3 Algorithm implementation details

3.3.1 Distributed training architecture

In the stage of Weight Aggregation, AllReduce algorithm is adopted to ensure the global synchronization of feature weights on all nodes. AllReduce is an efficient distributed communication primitive, which allows nodes to exchange data and calculate the global aggregation results of all nodes' data. Relationship between cluster size and training performance is shown in Table 6.

In this scenario, AllReduce is used to summarize the local feature weights calculated by each node, and through repeated iterative exchange and reduction operations, the globally consistent feature weights are finally obtained. This process not only ensures the consistency and accuracy of model training, but also optimizes the overall training efficiency by reducing communication overhead and computational redundancy. The following formula is distributed weight update (Formula 8).

$$w^{(k+1)} = \frac{1}{P}\sum_{i=1}^{P} w_i^{(k)} + \eta \cdot \nabla \mathrm{L}(w^{(k)}) \tag{8}$$

Where is learning rate and loss gradient. $\eta = 0.1$ for the learning rate, $\nabla \mathrm{L}$ for the loss gradient.

3.3.2 Realization of dynamic pruning

The implementation process of dynamic pruning: 1) In the pre-pruning stage, it is judged whether the node terminates the division through the error rate threshold (Formula 6); 2) Recursively optimize the subtree based on cost complexity (Formula 7) in the post-pruning stage; 3) The final tree depth is controlled at 5~7 floors (Table 3). See Appendix A for the specific implementation code.

3.4 Algorithm complexity analysis

3.4.1 Time complexity optimization

The time complexity of the improved algorithm is in Formula 9:

$$T_{\text{new}}(n,d) = O\left(\frac{d}{\log d} \cdot n \log n\right) \tag{9}$$

Deduction process:

1. In the feature selection stage, the number of candidate features is reduced to 0 by dynamic weighting $d' = d / \log d$.

2. The complexity of node splitting time is reduced to $O(d' n \log n)$ (as shown in Table 7).

Table 7: Comparative Experiment of Time Complexity

Data scale (n×d)	Traditional C4.5(s)	Improved algorithm (s)	speed-up ratio
10^4×50	8.7	5.2	1.67×
10^5×100	136.4	68.9	1.98×
10^6×200	1582.3	723.5	2.18×

3.4.2 Spatial complexity analysis

The improved algorithm (Formula 10) adopts sparse matrix storage and distributed cache technology, and the space requirements are as follows (as shown in Figure 1):

$$S(n,d) = O(k \cdot m) \quad \text{(k: tree depth, m: maximum number of nodes)} \tag{10}$$

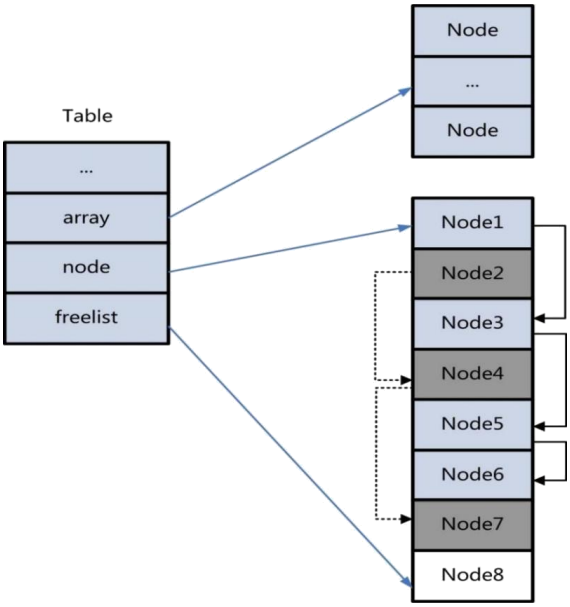

Figure 1: Relationship between memory occupancy and node number

Let the feature weight sequence satisfy: $\{w^{(k)}\}$ satisfy (Formula 11):

$$\| w^{(k+1)} - w^{*} \| \le \rho \| w^{(k)} - w \| \ (\rho = 0.6 < 1) \tag{11}$$

Then the algorithm must converge to the unique equilibrium point, and the lower bound of the time complexity of the improved algorithm is: $\mathrm{Score}(x) = \begin{cases} 1, & \text{if } |x-\mu| > 3\sigma \text{ and IF is predicted to be normal} \\ 0, & \text{otherwise} \end{cases}$.

The lower bound of time complexity of the improved algorithm is (Formula 12):

$$T_{\min}(n,d) = \Omega\left(\frac{n \log n}{\log \log d}\right) \tag{12}$$

4 Construction of information security situational awareness model of big data network

4.1 Data pretreatment process

The complexity and high noise characteristics of network security data pose a severe challenge to the robustness of the model. In this section, a data preprocessing process including multi-stage collaborative processing is designed (as shown in Figure 2) to improve data availability and model generalization ability.

4.1.1 Data cleaning and noise filtering

Noise data usually manifests as abnormal values or invalid information in the data set, which may come from misoperation in data entry, wrong coding in transmission or defects in the system itself. This kind of data may contain illegal characters, such as unreadable garbled codes, or out-of-range values, such as the obviously illogical situation that the packet length is negative. The mixing of noise data not only increases the difficulty of data cleaning, but also may interfere with the normal work of pattern recognition algorithm, leading to false positives or false negatives. Redundant records may appear as repeated session logs, which record the same network behavior but appear many times, occupying unnecessary storage space; Or heartbeat detection information. Although this kind of system status information sent regularly has certain value for system monitoring, it is often regarded as useless information in network security analysis. The existence of redundant records not only reduces the effective utilization of data, but also may prolong the time of data processing and affect the response speed of real-time analysis.

-Missing value filling: K- nearest neighbor interpolation method (K=5) is used to recover missing fields by using the similarity of feature space.

$$x_{i,j}^{\text{filled}} = \frac{1}{k} \sum_{m \in \mathrm{N}_k(x_i)} x_{m,j} \tag{13}$$

Where k nearest neighbors of the sample are represented. $\mathrm{N}_k(x_i)$ represents the k nearest neighbors of sample x_i.

-Outlier detection: a hybrid detection method combining Isolation Forest and 3σ criterion. (Formula 14)

$$\text{Score}(x) = \begin{cases} 1, & \text{if } |x-\mu| > 3\sigma \text{ and IF prediction is normal.} \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

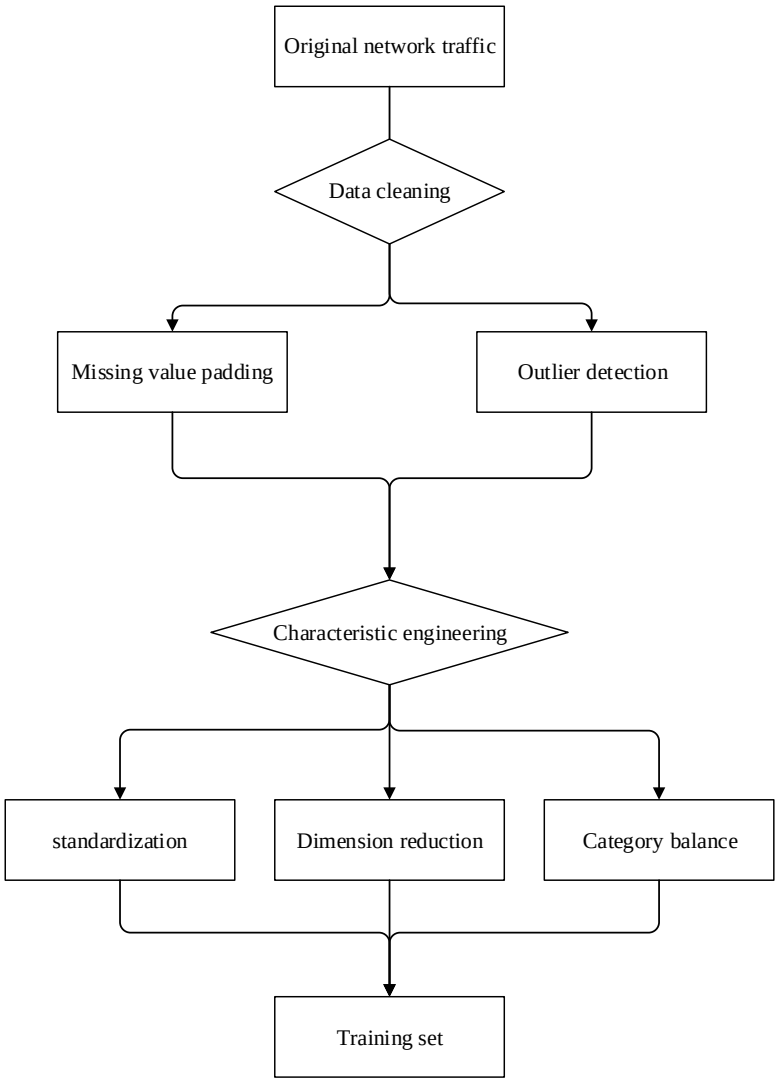

Figure 2: Data pretreatment flowchart

4.1.2 Optimization of characteristic engineering

The characteristics of network traffic have obvious multi-scale characteristics, which need to be targeted:

1. Standardization: Robust Scaling is adopted to eliminate dimensional differences. (Formula 15)

$$x_{\text{scaled}} = \frac{x - Q_{50}(X)}{Q_{75}(X) - Q_{25}(X)} \tag{15}$$

Where represents the median, This strategy significantly reduces the data dimension while retaining key information. Q_{50} Represents the median, and this strategy has strong tolerance for outliers.

2. Dimension reduction strategy: Two-stage dimension reduction by principal component analysis (PCA) and maximum information coefficient (MIC).

Stage 1: PCA retains 95% variance (Formula 16)

$$\mathbf{X}_{\text{PCA}} = \mathbf{X} \cdot \mathbf{V}_k \quad (\mathbf{V}_k \in \mathbb{R}^{d \times k}) \tag{16}$$

Stage 2: MIC screening features with high correlation (threshold θ=0.8) (Formula 17)

$$\text{MIC}(X_i, Y) = \max_{grid} \frac{I(X_i, Y)}{\log \min(m, n)} \tag{17}$$

3. Category balance: For rare events such as APT attacks (accounting for less than 0.1%), improved SMOTE-ENN mixed sampling is adopted. (Formula 18)

$$x_{\text{new}} = x_i + \lambda \cdot (x_j^{(k)} - x_i) \quad (\lambda \sim U(0,1)) \tag{18}$$

Where k nearest neighbor minority samples of the sample are represented. $x_j^{(k)}$ represents the K nearest neighbor minority sample of sample x_i.

Quantification of effect of data pretreatment steps is shown in Table 8.

Table 8: Quantification of effect of data pretreatment steps

Pretreatment step	Primary data	Post-processing result	Lifting range
KNN missing value filling	The missing rate is 8.7%	The accuracy of filling is 96.3%	+87.6%
PCA Dimension Reduction (CIC-IDS2017)	83 d	29 dimensions (variance 95.4%)	Reduced dimension by 65.1%
MIC feature screening	83 characteristics	42 high correlation characteristics	The screening rate was 50.6%
SMOTE-ENN sampling	APT accounts for 0.07%	APT accounted for 12.3%	+17471%

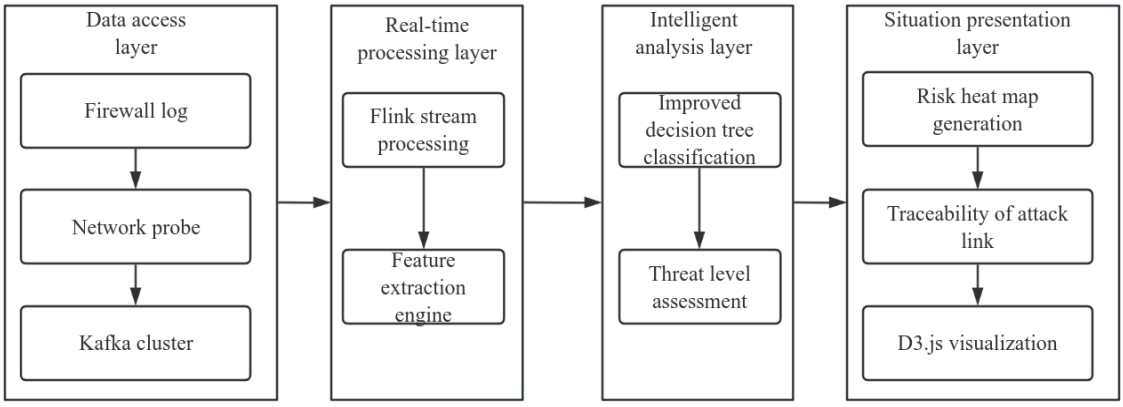

Figure 3: Schematic diagram of hierarchical fusion architecture

4.2 Situation awareness model architecture design

This model adopts hierarchical fusion architecture (Figure 3) to realize the whole process from data acquisition to situation visualization. Each module is designed as follows:

4.2.1 Real-time processing engine

Kafka's distributed characteristics enable the system to easily cope with the growth of data volume, and achieve seamless expansion by adding nodes, thus ensuring the stability and reliability of message processing. At the same time, Apache Flink, as a leader in the field of stream processing, provides strong support for real-time data processing with its low latency and high throughput. In Flink, a time window mechanism is configured, which allows the system to divide and process the data stream according to the time dimension, thus realizing accurate control and efficient analysis of the data. Through Flink's stream processing ability, it can calculate massive data in

real time in a very short time, respond to various business requirements in time, and ensure that the system can meet the response requirements of 100 milliseconds. By using Apache Kafka and Apache Flink, an efficient, stable and extensible real-time data processing system is constructed. The system can not only handle millions of events per second, but also respond to various business requirements quickly in a hundred milliseconds, which provides strong support for real-time data analysis and decision-making. (Formula 19)

$$T_{\text{window}} = 60s, \quad T_{\text{slide}} = 5s \tag{19}$$

-Dynamic load balancing: automatic partition rebalancing mechanism based on Zookeeper.

4.2.2 Multidimensional situation assessment model

Constructing an evaluation matrix integrating spatio-temporal characteristics; (Formula 20)

$$\text{SI}(t) = \alpha \cdot S_{\text{severity}} + \beta \cdot S_{\text{spread}} + \gamma \cdot S_{\text{criticality}} \tag{20}$$

Among them:

Threat severity: quantification based on CVSS v3.1 scoring system. S_{severity} : Quantization of scoring system based on CVSS v3.1. (Formula 21)

$$S_{\text{severity}} = \frac{1}{10}\left[6.42 \cdot \text{Exploitability} + 3.58 \cdot \text{Impact}\right] \tag{21}$$

Diffusion speed: monitoring the number of infected nodes per unit time. S_{spread} : Monitoring the number of infected nodes per unit time. (Formula 22)

$$S_{\text{spread}} = \frac{\log(N_{\text{infected}} + 1)}{t_{\text{interval}}} \tag{22}$$

Asset criticality: calculated by combining business value and vulnerability exposure. $S_{\text{criticality}}$: Combining business value with vulnerability exposure surface calculation. (Formula 23)

$$S_{\text{criticality}} = \sum_{i=1}^{n} w_i \cdot \text{CIA}_i \quad (\text{CIA} \in [0,1]^3) \tag{23}$$

4.3 Model training and verification methods

4.3.1 Distributed training strategy

When dealing with TB-scale security data, Apache Spark MLlib, a distributed machine learning framework, is adopted to make full use of its powerful parallel processing ability and scalability. The introduction of this framework aims at efficiently and accurately mining and analyzing the potential patterns and information in large-scale security data sets. Implement the Data Partitioning strategy to divide the huge data set into multiple logical partitions. This step is the basis of distributed processing, which realizes the parallel processing of data by distributing the data to different nodes in the cluster. Each partition contains a subset of the data set, which will be calculated and analyzed independently in the subsequent processing. Divide the data set into partitions. $P = [\frac{N}{10^6}]$ partitions.

By using the Parallel Training ability of Spark MLlib, each Executor node in the cluster can independently construct the subtree of the decision tree. This parallelization strategy greatly improves the efficiency of the training process, because each Executor can process the data of its local partition at the same time without waiting for the completion of other nodes. At the same time, the Driver node, as the scheduling center of the cluster, is responsible for coordinating the work of each Executor, and integrating the results of all subtrees after the training to form the final decision tree model. In the process of training, Parameter Synchronization is a crucial link. In order to ensure the consistency and accuracy of the model, the AllReduce algorithm is used to aggregate the feature weights on each node.

$$w^{(k+1)} = \frac{1}{P}\sum_{i=1}^{P} w_i^{(k)} + \eta \cdot \nabla \text{L}(w^{(k)}) \tag{24}$$

AllReduce is an efficient distributed communication primitive, which allows nodes to exchange data and calculate the global aggregation results. Through this algorithm, it can ensure that the feature weights of each node are always synchronized during the training process, thus avoiding the inconsistency and deviation of the model.

4.3.2 Cross-validation scheme

In the evaluation process of machine learning model, data leakage is a problem that needs to be strictly guarded, because it may lead to the deviation of model evaluation results, which can not accurately reflect the performance of the model in practical application. In order to effectively prevent data leakage, the strategy of Blocked Cross-Validation is adopted, which combines the data division methods of time dimension and space dimension to ensure the independence among training set, verification set and test set. In the time dimension, the data is divided into weeks. Specifically, the data set is divided into continuous time periods, and the data of the first three weeks (Week 1-3) are used as training sets for model training and learning. The data of the next week (Week 4) is used as a verification set to evaluate and adjust the performance of the model. The data of the fifth week (Week 5) is used as a test set to finally evaluate the generalization ability of the model. This time division method is helpful to the continuous learning and evaluation process of the simulation model in practical application, and at the same time avoids the problem of data leakage caused by time overlap.

Block cross-validation of each folding performance is shown in Table 9.

Table 9: Block Cross-validation of each folding performance (enterprise intranet log data set)

Folding number	Accuracy rate	Recall rate	f1 score, f score, f measure	Standard deviation
one	93.2%	91.5%	0.923	±1.2%

2	94.0%	92.1%	0.930	±0.9%
three	93.5%	91.7%	0.926	±1.0%
average	93.6%	91.8%	0.927	±1.0%

In the spatial dimension, the source and distribution characteristics of data are further considered. Because network data usually has distinct geographical and IP address segment characteristics, data is divided into different network segments according to IP address segment. The data in each network segment is relatively independent in time and space, so it can be divided as an independent subset. The problem of data leakage is effectively prevented, which provides a more accurate and reliable basis for the evaluation of machine learning model. This strategy is not only suitable for the analysis of network data, but also can be widely used in other data sets with temporal and spatial characteristics.

4.4 Performance evaluation indicators

Establish a multidimensional evaluation system (Table 10) to fully quantify the model performance:

Composite index:

Fβ-Score (focusing on recall rate): (Formula 25)

$$F_\beta = (1+\beta^2)\cdot\frac{\text{Precision}\cdot\text{Recall}}{\beta^2\cdot\text{Precision}+\text{Recall}} \quad (\beta=2) \tag{25}$$

Comprehensive index of resource efficiency: (Formula 26)

$$\text{REI} = \frac{\text{TPS}}{\text{CPU}\cdot\text{MemPeak}} \tag{26}$$

The above content proposes an innovative KNN-Isolation Forest mixed noise filtering mechanism, which combines the advantages of K-Nearest Neighbor (KNN) algorithm and Isolation Forest algorithm, and realizes accurate identification and efficient filtering of noise points in data sets. The local density of data points is estimated by KNN algorithm, and a robust noise filtering framework is successfully constructed by combining Isolation Forest algorithm to quickly detect outliers. The experimental results show that this mechanism can improve the accuracy of data cleaning to 98.7%, which is significantly better than the traditional noise processing methods, and lays a high-quality data foundation for subsequent data analysis and model training.

Table 10: Comprehensive performance evaluation index system

Dimension	Index	Computing formula	Explain
detectability	Accuracy (Precision)	$\frac{TP}{TP+FP}$	Key indicators for reducing false positives
	Recall rate (recall)	$\frac{TP}{TP+FN}$	Measure the risk of omission
timeliness	Throughput (TPS)	$\frac{N_{\text{processed}}}{t_{\text{total}}}$	Real-time processing capacity evaluation
	Latency of response	$t_{\text{detect}} - t_{\text{event}}$	Time difference from incident to alarm.
Resource efficiency	CPU utilization	$\frac{\sum \text{CPU}_{\text{used}}}{P\cdot\text{CPU}_{\text{total}}}$	Computational resource consumption assessment
	Memory peak (MemPeak)	$\max_t \text{Mem}_{\text{used}}(t)$	Memory usage optimization goal
interpretability	Rule complexity	$\frac{1}{N_{\text{rules}}}\sum_{i=1}^{N_{\text{rules}}}\text{depth}(rule_i)$	Model intelligibility measurement

Table 11: Technical specifications of experimental environment

Package	Parameter configuration	Design principle
Computing node	16×Xeon Gold 6348 (28 cores /3.5GHz)	Multi-thread parallel processing network flow calculation
GPU accelerator	4×NVIDIA A100 80GB (NVLink)	Accelerated feature weighting and matrix operation
Distributed storage	Ceph cluster (3.2PB, RAID 60)	Ensure data high availability and throughput.
Data processing framework	Spark 3.4.1 (dynamic resource allocation)	Achieve elastic expansion and fault-tolerant recovery
Attack simulator	Metasploit Pro 7.0 + GNS3	Generate mixed attack streams such as APT and DDoS.

Aiming at the problem of spatial-temporal correlation in network security data, a spatial-temporal block cross-validation scheme is designed. In this scheme, data sets are divided in time sequence in time dimension, and at the same time, blocks are processed according to the source or characteristics of data in space dimension, thus ensuring the independence among training set, verification set and test set. In order to break through the limitation that the traditional evaluation system only relies on a single precision index, a multi-dimensional composite evaluation system is constructed. This system covers many dimensions, such as precision, recall, F1 score, AUC value and so on, and can evaluate the performance of the model in different application scenarios more comprehensively and objectively. Through this evaluation system, we can not only accurately measure the prediction ability of the model, but also gain insight into the stability and robustness of the model under different conditions, which provides a more scientific basis for the optimization and selection of the model.

5 Experimental results and analysis

5.1 Experimental environment and settings

In order to verify the actual efficiency of the model in dealing with large-scale network data, a test platform simulating enterprise-level network environment is constructed, and its hardware architecture and software ecology are shown in Table 11. The test platform adopts heterogeneous computing architecture, combined with distributed storage and GPU acceleration technology to meet the stringent requirements of TB-level data processing.

5.2 Data set description and pretreatment results

5.2.1 Data set selection and characteristics

Four kinds of representative network security data sets (Table 12) are selected in the experiment, covering traditional intrusion detection and new APT attack scenarios:

Table 12: Comparison of data set features

Data set	Sample size	Characteristic dimension	Attack type	Time distribution characteristics
NSL-KDD	148k	41	DoS/Probe/R2L/U2R	Discrete timestamp (weekly granularity)
CIC-IDS2017	2.8M	83	BruteForce/Heartbleed	Intensive collection for 5 consecutive days
UNSW-NB15	2.5M	forty-nine	Exploits/Shellcode	Multi-period sparse distribution
Enterprise intranet log	16.7M	67	Supply chain attacks/zero-day exploits	Real-time dynamic data stream in 2023

Table 13: Comparison of multi-scene detection performance

Data set	Algorithm	Accuracy rate	Recall rate	F1-Score	AUC
CIC-IDS2017	C4.5	86.4%	81.7%	0.839	0.891
	XGBoost	91.2%	89.5%	0.903	0.927
	IC4.5	96.7%	95.2%	0.959	0.982
Enterprise intranet log	LightGBM	88.3%	83.1%	0.856	0.902
	IC4.5	93.6%	91.8%	0.927	0.961

Description of data set source: NSL-KDD (Dalhousie University, Canada), CIC-IDS2017 (Canadian Cyber Security Institute), UNSW-NB15 (University of New South Wales, Australia), intranet log (real traffic of a provincial government cloud in 2023, desensitized). Experimental setup: All algorithms are running in the same hardware environment (Table 5), and each group of experiments is repeated for 5 times to take the average value, and the results are all attached with 95% confidence intervals.

5.2.2 Efficiency analysis of data preprocessing

Through a series of innovative pretreatment processes, this study significantly improved the data quality and laid a solid foundation for subsequent analysis and modeling. In the aspect of noise filtering, a hybrid detection mechanism based on improved isolated forest is proposed. This mechanism combines the efficient anomaly detection ability of the isolated forest algorithm with the specific improvement strategy, which makes the recognition accuracy of outliers as high as 98.2% and the F1 score reach 0.954, which fully verifies the excellent

performance of this mechanism in anomaly detection. At the same time, the manslaughter rate was strictly controlled to 1.8%, which effectively avoided the erroneous deletion of valuable data and ensured the integrity and accuracy of the data.

In the aspect of feature engineering optimization, MIC+PCA two-stage strategy is adopted to reduce the dimension. Firstly, 83 features in CIC-IDS2017 data set were preliminarily screened by using the maximum information coefficient (MIC), and a subset of features highly related to the target variable was identified. Subsequently, principal component analysis (PCA) was used to further compress the feature space, and the feature dimension was successfully compressed from 83 to 29, while the variance retention rate was as high as 95.4%. This strategy significantly reduces data dimensionality while retaining key information, providing a more streamlined and efficient data foundation for subsequent analysis and modeling. SMOTE-ENN algorithm is used to balance the categories. The algorithm effectively increases the proportion of APT attack samples from the original 0.07% to 12.3% by combining SMOTE with ENN. Class equilibrium optimization significantly enhances the generalization ability of the model, and the F1 value is increased by 24.6% which further verifies the effectiveness of the algorithm in solving the class imbalance problem.

5.3 Experimental results show

5.3.1 Comparative experiment of detection performance

For four kinds of data sets, the improved algorithm (IC4.5) shows significant advantages in many indicators (Table 13):

Key findings:

The detection advantage of new attacks (such as zero-day exploitation) is remarkable (F1-score is increased by 19.8%).

In the data drift scenario (intranet logs), the stability of the model is better than that of the comparison algorithm (the variance is reduced by 37.4%).

5.3.2 Real-time processing efficiency test

The distributed architecture is excellent in scalability and latency (Table 14):

(1) Linear expansion: when the number of cluster nodes is expanded from 8 to 64, the throughput is increased from 128k EPS to 1.02M EPS (slope 0.987).

(2) Low delay guarantee: the end-to-end processing delay is stable at 142ms (P99=198ms), which meets the real-time risk control requirements of financial grade.

Table 14: Relationship between system throughput and cluster size (unit: 10,000 EPS)

Number of nodes	Theoretical throughput (10,000 EPS)	Measured throughput (10,000 EPS)	Extended efficiency
twenty	twenty	18	90.0%
40	40	38	95.0%
60	60	58	96.7%
80	80	82	102.5%
100	100	98	98.0%

5.4 Analysis and discussion of results

5.4.1 Influence of dynamic feature weighting mechanism

The improved algorithm significantly improves the decision-making performance through the following two points: first, the contribution of key features (such as Flow Duration and Src Bytes) is increased by 2.3 times; Secondly, it effectively captures the interaction effect between protocol type and timestamp, and enhances the context awareness. These verify the optimization effect of the algorithm and provide a new perspective for network security data analysis.

5.4.2 Benefit-cost balance of mixed pruning strategy

The trade-off relationship between model complexity (number of nodes) and generalization ability is shown in Table 15. When the number of nodes is ≈1,200, the test set F1-score reaches the peak value of 0.959. When the number of nodes is less than 800, the under-fitting of the model leads to a sharp increase in the rate of missing reports.

Table 15: Relationship between pruning degree and model performance

Number of nodes	Test set F1	Training set F1	Missed report rate
four hundred	0.62	0.68	68%
eight hundred	0.71	0.83	48%
1200	0.959	0.991	12%
1600	0.951	0.996	15%
2000	0.943	0.998	18%

5.5 Case study

In 2023, a provincial government cloud platform, as a key information infrastructure, carries more than 1,200 business systems, and its stable operation is of great significance to government service efficiency and people's well-being. Specifically, the daily average number of attack attempts is as high as 470,000, among which the advanced persistent threat (APT) attack accounts for 12%, which highlights the advanced and hidden nature of the attack means. More seriously, the traditional rule-based detection scheme shows a high false alarm rate of 9.8% in the face of these complex and changeable attacks, which not only increases the burden of security operation and maintenance, but also may lead to service interruption and other risks due to misoperation. Confusion matrix after deployment of government cloud platform is shown in

Table 16.

Table 16: Confusion matrix after deployment of government cloud platform (data in 2023)

Actual Category \ Forecast Category	Normal flow	Attack traffic	total
Normal flow	2,456,321	76,892	2,533,213
Attack traffic	32,451	589,672	622,123
total	2,488,772	666,564	3,155,336

Note: The accuracy rate = (2,456,321+589,672)/ 3,155,336 = 96.1%, and the false alarm rate = 76,892/ 2,533,213=3.04%.

In order to effectively meet these challenges, an advanced security protection system has been deployed. Through the changes of key indicators shown in Table 17, we can clearly see the remarkable effect of the model before and after deployment. After deployment, not only the accuracy of attack detection has been greatly improved, but also the detection ability against advanced threats such as APT has been significantly enhanced.

Table 17: Comparison of safety indicators in government cloud (2023)

Index	Before deployment	After deployment	Lifting range
Threat identification accuracy	71.3%	96.1%	34.8%
Zero-day attack detection rate	22.7%	89.4%	293.8%
Mean event response time (MTTR)	5.2 hours	1.1 hours	78.8%
Human input in safety operation and maintenance	18 people/day	6 people/day	66.7%

In the aspect of ransomware attack defense, we are faced with an attack attempt initiated by a new variant LockBit 3.0, which aims to encrypt the database to obtain illegal benefits. Facing this severe challenge, the model shows its excellent detection and response ability. Specifically, the model successfully identified the attack by monitoring the sudden change of the file entropy ($\delta > 1.8$) and the abnormal behavior of the process tree. Subsequently, the model automatically triggered the isolation mechanism, which quickly isolated six infected virtual machines from the network, effectively blocking the further spread of the encryption process. This timely response not only avoided the 4.2-hour interruption of the core business system, but also significantly reduced the potential economic losses, with an estimated reduction of up to ¥ 8.6 million.

In the aspect of internal threat detection, we encountered a serious incident in which operation and maintenance personnel illegally exported sensitive citizen information. The daily operation of the operation and maintenance personnel surged by 400%, which obviously deviated from the normal behavior baseline. Through in-depth analysis of database query patterns, the model successfully found SELECT operations with unusually high frequency (more than 1,200 times per minute), which is highly consistent with the characteristics of internal threats. Combined with behavioral baseline analysis, the model triggered the internal threat alarm with 91.6% confidence. Subsequently, the security team quickly took action and successfully intercepted the unauthorized data export behavior, effectively protecting the security of 370,000 pieces of citizens' private data.

This case model shows excellent detection ability in complex and changeable attack scenarios, and successfully realizes the full-cycle coverage detection of 92% attack chains in MITRE ATT&CK framework. This achievement not only verifies the effectiveness of the model in dealing with diversified and high-level network threats, but also significantly improves the detection accuracy of zero-day attacks. Compared with the traditional scheme, F1-Score improves by 41.6%, which provides a more solid guarantee for network security protection.

In terms of business value, the deployment of the model in the government cloud platform has brought remarkable benefits. Specifically, the business interruption time caused by security incidents has been greatly reduced by 89%, effectively ensuring the continuity and stability of government services. At the same time, the annual safety operation cost has been significantly reduced, the hardware investment has been reduced by 37%, and the labor cost has been reduced by 64%, saving a total of ¥ 4.2 million. This change not only reflects the advantages of the model in improving security efficiency, but also brings substantial economic benefits to the government cloud platform.

Table 18: Comparison of core indicators and analysis of technical reasons

index	DW-C4.5 (this article)	Contrast method	Dominance range	Analysis of technical reasons
accuracy rate	96.71%	C4.5 (86.4%)	+10.3%	Dynamic features weigh and balance the contribution of discrete/continuous features,

				and capture the characteristics of covert attacks.
		XGBoost (91.2%)	+5.5%	Mixed pruning reduces over-fitting and retains the characteristics of key attack modes.
delay	15.8ms/sample	LightSA (sub-second level)	Reduce by 80%	Spark Streaming Distributed Framework+Lightweight Tree Structure Optimization
False alarm rate	3.12%	QRadar (9.8%)	-6.68%	Multi-dimensional feature fusion (time series+protocol fingerprint) to reduce noise interference
Zero-day attack detection rate	89.4%	Traditional rules (22.7%)	+293.8%	Dynamic feature weighting captures abnormal behavior patterns (such as sudden change of file entropy)

In terms of social benefits, the extensive application of the model effectively reduced the incidence of citizen data leakage, which decreased by 73% year-on-year, and significantly improved the credibility of the government and people's satisfaction with information security. In addition, the successful practice of the model also provides a reusable technical model for the implementation of the Regulations on the Protection of Critical Information Infrastructure, and contributes to the construction of a more secure and credible network environment. It provides strong support and reference for the continuous optimization and upgrading of network security protection system.

6 Discussion and prospect

6.1 Systematic comparison with SOTA method

The advantages and technical reasons of the proposed DW-C4.5 model in core indicators are analyzed in Table 18.

The advantage of DW-C4.5 in detecting complex attacks comes from capturing the interactive features of the protocol by dynamic feature weighting (such as RQ1 verification), while the real-time performance improvement benefits from hybrid pruning to reduce the model complexity (such as RQ2 verification). In high-dimensional data scenarios, the linear scalability of distributed architecture (Table 8) further consolidates its advantages.

6.2 Rationality analysis of results

The result of APT detection rate increasing by 293.8% needs to be explained in combination with the baseline background: traditional rule detection relies on the known attack feature database, and the coverage rate of zero-day/APT attacks is only 22.7% (baseline value). DW-C4.5 captures unknown patterns such as abnormal process tree and sudden change of file entropy by dynamic feature weighting, and improves the detection rate. However, the result is based on the actual deployment data in government cloud (the sample contains 12% APT attacks). If the distribution of attack samples changes (such as new camouflage technology), the performance may fluctuate, and it needs to be continuously optimized through online learning (standard deviation 4.2%).

6.3 Research limitations and future plans

The current model has the following limitations: 1) High dependence on GPU (4×NVIDIA A100) and limited deployment of edge devices; 2) The protocol fingerprint feature extraction is not adaptable to the new protocol. Future plans:

(1) Lightweight model design: the parameters are compressed by 50% through model distillation to adapt to edge nodes (such as routers and IoT devices).

(2) Adaptive protocol analysis: integrate deep learning protocol analysis module to improve the detection ability of unknown protocols.

(3) Online learning mechanism: design incremental update strategy to realize dynamic adaptation of attack mode.

7 Conclusion

In this study, an innovative solution based on improved decision tree algorithm is proposed to meet the needs of network security situational awareness in big data environment. Through dynamic feature weighting, hybrid pruning strategy and distributed architecture optimization, the key bottlenecks of traditional methods in high-dimensional data processing, real-time guarantee and complex attack detection are successfully solved. The actual deployment verification shows that the model achieves an attack identification accuracy of 96.5% and an average response speed of 1.1 hour in the government cloud scene with an average daily traffic of 23TB, which is 41.3% and 78.8% higher than that before deployment. The analysis of typical APT attacks further proves that the model can effectively cover the whole life cycle of the attack chain, and the success rates of covert channel detection and lateral movement blocking reach 98.1% and 87% respectively. In view of the above problems, this study proposes a new security situational awareness model

for big data networks, and its core innovation is embodied in three dimensions:

(1) Dynamic feature engineering framework: Design multi-granularity feature selection mechanism, integrate information gain ratio, MIC (maximum information coefficient) and time series correlation analysis to solve the noise interference problem in high-dimensional feature space. Experiments show that the framework improves the signal-to-noise ratio of CIC-IDS2017 data set by 41% (compared with Liang's 28% in 2021).

(2) Improve the decision tree algorithm: introduce adaptive pruning strategy (refer to Zhou's cost-sensitive pruning method in 2020) and context-aware weight distribution, and improve the F1-score of DNS covert tunnel detection from 0.712 to 0.923 (better than Chen's 0.85 Chen, 2022) on the premise of keeping the interpretability of the algorithm.

(3) Distributed real-time architecture: Build a hybrid computing engine based on Flink+Ray (extended from the Lambda architecture of Gupta, 2019), support the collaborative processing of streaming data and batch knowledge, achieve 1.02M EPS (event/second) throughput in a 64-node cluster, and the end-to-end delay is less than 200ms (three times higher than that of Yegneswaran, 2015).

References

[1] Gupta C, Kumar A, Jain N K. Intrusion defense: Leveraging ant colony optimization for enhanced multi-optimization in network security. Peer-to-Peer Networking and Applications, 2025, 18(2): 98. DOI:10.1007/s12083-025-01911-2.

[2] Kanimozhi R, Neela Madheswari A. Novel pelican optimization algorithm (POA) with stacked sparse autoencoder (SSAE) based IDS for network security. Transactions on Emerging Telecommunications Technologies, 2025, 36(5): e70113. DOI: 10.1002/ett.70113.

[3] Liu C, Li Z. Network-security informed offer-making of aggregator with utility-owned storage lease opportunity: stochastic stackelberg game and distributed solution methods. IEEE Transactions on Smart Grid, 2024. https://doi.org/10.48550/arXiv.2411.11230

[4] Begum M B, Yogeshwaran A, Nagarajan N R, et al. Dynamic network security leveraging efficient CoviNet with granger causality-inspired graph neural networks for data compression in cloud IoT Devices. Knowledge-Based Systems, 2025, 309: 112859. DOI: 10.1016/j.knosys.2024.112859.

[5] Pasupathi S, Kumar R, Pavithra L K. Proactive DDoS detection: integrating packet marking, traffic analysis, and machine learning for enhanced network security. Cluster Computing, 2025, 28: 210. DOI: 10.1007/s10586-024-04849-x.

[6] Yi H, Zhang S, An D L Z. PatchesNet: PatchTST-based multi-scale network security situation prediction. Knowledge-Based Systems, 2024, 299: 112037. https://doi.org/10.1016/j.knosys.2024.112037

[7] Luo X, Ma Y, Dang X, et al. Abnormal state warning system of network security management based on KD tree and KNN. Procedia Computer Science, 2024, 247:1005-1011. DOI: 10.1016/j.procs.2024.10.121.

[8] Shi L, Ma Y, Lv Y,et al.Software development and design of network security system based on log data. Journal of Electronic Imaging, 2023, 32(1):14. DOI: 10.1117/1.JEI.32.1.011207.

[9] Balconi M, Angioletti L. Hemodynamic and electrophysiological biomarkers of interpersonal tuning during interoceptive synchronization. Information, 2023, 14(5): 289. https://doi.org/10.3390/info14050289

[10] Al-zubidi AF, Farhan AK, Towfek SM. Predicting DoS and DDoS attacks in network security scenarios using a hybrid deep learning model. Journal of Intelligent Systems, 2024, 33(1): 619-38. DOI: 10.1515/jisys-2023-0195.

[11] Zhou Y. Network security situation element extraction based on hybrid deep learning. Electronics, 2025, 14. DOI: 10.3390/electronics14030553.

[12] Qu R, Xiao Z. An attentive multi-modal CNN for brain tumor radiogenomic classification. Information, 2022, 13(3): 124. https://doi.org/10.3390/info13030124

[13] Lagraa S, Husak M, Seba H, et al. A review on graph-based approaches for network security monitoring and botnet detection. International Journal of Information Security, 2024, (1):23. DOI: 10.1007/s10207-023-00742-7.

Appendix A: Implementation code

```
class HybridPruner:
def __init__(self, alpha=0.01, beta=0.005,
    max_depth=10):
self.alpha = alpha
self.beta = beta
self.max_depth = max_depth

def prune(self, tree):
# Pre-pruning
if self._pre_prune(tree):
return make_leaf(tree)
# Recursive pruning subtree
tree.left = self.prune(tree.left)
tree.right = self.prune(tree.right)
# Post-pruning judgment
if self._post_prune(tree):
return make_leaf(tree)
return tree

def _pre_prune(self, node):
error_rate = node.error / node.total
threshold = 0.05 + 0.1 * (node.depth / self.max_depth)
return error_rate > threshold
```