

ITU-T Technical Report

(12/2025)

XSTR.se-AI

Security evaluation on artificial intelligence technology in information and communication technology

Technical Report ITU-T XSTR.se-AI

Security evaluation on artificial intelligence technology in information and communication technology

Summary

Evaluating the security of artificial intelligence (AI) technology is a crucial step in achieving technological optimization and risk management, thereby promoting the development of AI technology towards a more mature and responsible trajectory. The evaluation of AI security encompasses a range of indicators. However, there is no unified standard for an AI security evaluation matrix and clear evaluation methods, which might cause a lack of objectivity, effectiveness and operability in application.

Technical Report ITU-T XSTR.se-AI presents an AI security evaluation framework including evaluation dimensions (data, model, and environment) and evaluation indicators for each evaluation dimension. It also tries to give related evaluation methods for data, models, and environments, providing conclusive evidence and reference for evaluating the security of AI technology.

This Technical Report can assist AI practitioners in identifying existing issues, making improvements, and determining the most suitable scenarios for AI technology. AI practitioners can use the knowledge in this report to evaluate the features and levels of certain security categories or indicators of a certain component of AI technology or system.

Keywords

Data evaluation, environment evaluation, model evaluation, security evaluation of AI, security indicator.

Note

This is an informative ITU-T publication. Mandatory provisions, such as those found in ITU-T Recommendations, are outside the scope of this publication. This publication should only be referenced bibliographically in ITU-T Recommendations.

© ITU 2026

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

Table of Contents

	Page
1 Scope.....	1
2 Reference	1
3 Definitions	1
3.1 Terms defined elsewhere	1
3.2 Terms defined in this Technical Report	1
4 Abbreviations and acronyms	1
5 Conventions	2
5 AI key security indicator analysis.....	2
7 AI security evaluation methods	7
7.1 AI data security evaluation methods	8
7.2 AI model security evaluation methods	16
7.3 AI environment security evaluation methods.....	20
7.4 Comprehensive evaluation methods.....	24
7.5 Conclusion.....	25
Annex A – Application use cases of AI security evaluation methods	26
A.1 Data security evaluation for intrusion detection.....	26
A.2 Model security evaluation for intrusion detection.....	29
A.3 Environment security evaluation for intrusion detection	31
A.4 Comprehensive evaluation for intrusion detection system.....	35
Bibliography.....	36

Security evaluation on artificial intelligence technology in information and communication technology

1 Scope

The Technical Report covers:

- Summarization of key security indicators of artificial intelligence (AI) technology from three perspectives: data, model, and environment;
- An AI security evaluation framework including evaluation (sub-)categories, security evaluation items, and evaluation methods for data, models, and environments separately; and
- Application of AI security evaluation methods in telecommunication scenarios.

2 Reference

None.

3 Definitions

3.1 Terms defined elsewhere

This Technical Report uses the following term defined elsewhere:

3.1.1 domain [b-ITU-T Y.3090]: A collection of physical or functional entities which are owned and operated by a player and can include entities from more than one role. The extent of a domain is defined by a useful context, and one player can have more than one domain.

3.2 Terms defined in this Technical Report

None.

4 Abbreviations and acronyms

This Technical Report uses the following abbreviations and acronyms:

AAB	Adversarial Accuracy for Black Box Attacks
AAW	Adversarial Accuracy for White Box Attacks
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
CII	Critical Information Infrastructure
CR	Consistency Ratio
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DDoS	Distributed Denial of Service attack
DeepLIFT	Deep Learning Important Features
FAHP	Fuzzy Analytic Hierarchy Process
FN	False Negative
FP	False Positive

FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ICMP	Internet Control Message Protocol
ICT	Information and Communication Technology
IDS	Intrusion Detection System
IoT	Internet of Things
IP	Internet Protocol
LIME	Local Interpretable Model-Agnostic Explanations
RI	Random consistency Index
ROC	Receiver Operating Characteristic
SDA	Stacked Denoising Autoencoder
SHAP	Shapley Additive explanations
SOS	Stochastic Outlier Selection
SQL	Structured Query Language
SVM	Support Vector Machine
TCP	Transfer Control Protocol
TEE	Trusted Execution Environment
TN	True Negative
TP	True Positive
UDP	User Datagram Protocol
VPN	Virtual Private Network

5 Conventions

None.

5 AI key security indicator analysis

AI technology is increasingly widely used in information and communication technology (ICT). From data analysis to automated decision-making, AI technology is changing our work and lifestyle. However, with the in-depth application of AI technology, its security issues are increasingly prominent. Therefore, it is very important to evaluate the security of AI technology.

This manuscript refers to the ITU Guidelines for security management using artificial intelligence technology [b-XSTR-SEC-AI] and the artificial intelligence risk management framework [b-NIST AI RMF 1.0] to form the analysis dimensions and specific indicators.

[b-XSTR-SEC-AI] divides the security technology of AI into four directions: service, algorithm, data and platform from the perspective of security threat analysis. From the point of view of specific technical content, services and platforms all belong to providing support for model operation, and specific technologies have a large overlap. Algorithm security technology includes model robustness evaluation, model interpretability evaluation, and other directions of model security technology. Thus, we combine services and platforms into environments and change algorithms into models to include algorithm security technology and model security technology, so that the security evaluation

of AI technology in ICT can be summarised into the three key dimensions of data, models and environments. Data includes data security in the training and inference phases of AI systems, such as data leakage, anti-sample attacks and other security risks; models include model security in the development, deployment and application phases of models, such as security risks caused by model extraction, model tampering, weak robustness and weak interpretability of models and environments include basic platform security, business service security, tool security, etc. The main goal of conducting an AI technology security evaluation is to maintain the confidentiality, completeness, and availability of AI systems, data, and related infrastructure.

[b-NIST AI RMF 1.0] lists seven features that a trustworthy AI system should have. We have filtered out the five security-related features from these: valid and reliable, secure and resilient, explainable and interpretable, privacy enhanced and fair.

Table 1 describes the definition of each feature and its association with security, as well as the AI evaluation indicators corresponding to each feature in the three dimensions of data, model and environment.

Table 1 – The security-related features in [b-NIST AI RMF 1.0] and corresponding AI evaluation indicators

Features	Definition	Degree of relevance to AI security	Data indicator	Model indicator	Environment indicator	Relevance of AI features and indicator
Valid and reliable	Validation is the "confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled" [b-ISO 9000]. Reliability is defined in the same standard as the "ability of an item to perform as required, without failure, for a given time interval, under given conditions" [b-ISO/IEC TS 5723].	Attackers often create human failures for AI systems to carry out system attacks, such as power outages or incorrect system outputs resulting from inputting incorrect data, so validity and reliability in AI systems are the basis for defending against attacks.	D_Accuracy, D_Completeness, D_Consistency	M_Accuracy, M_Credibility	E_Completeness	Valid and reliable implies that the model output must be accurate and credible, while the accuracy and credibility of the model are influenced by the data, which must also be accurate, complete and consistent to avoid inaccuracies, missing data and training data with inconsistent content or format affecting the accuracy and credibility of the model.
Secure and resilient	Security includes resilience but also encompasses protocols to avoid, protect against, respond to, or recover from attacks. [b-NIST AI RMF 1.0] AI systems, as well as the ecosystems in which they are deployed, may be said to be resilient if they can withstand unexpected adverse events or unexpected changes in their environment or use – or if they can maintain their functions and structure in the face of internal and external change and degrade safely and gracefully when this is necessary [b-NIST AI RMF 1.0].	Secure and resilience refer to the ability of the system to withstand attack or disruption and are influenced by other security-related characteristics.	D_Diversity, D_Balance	M_Robustness, M_Credibility	E_Vulnerability E_Robustness	Secure and resilient requires the system to be able to respond effectively to attacks, accidents and perturbations, both from the data aspect, by ensuring the diversity and balance of the training data to increase the generalisation ability of the model to defend against all kinds of accidents or noise in the environment. At the model aspect, robustness must be ensured so that the model can still produce correct results in the face of attacks or minor perturbations. In addition, credibility is the degree of confidence in the output results of the AI model.

Table 1 – The security-related features in [b-NIST AI RMF 1.0] and corresponding AI evaluation indicators

Features	Definition	Degree of relevance to AI security	Data indicator	Model indicator	Environment indicator	Relevance of AI features and indicator
						When the external attack increases the model's cognitive uncertainty, the credibility will be reduced. The user is then alerted that the model may present security risks at that time and that decision-making may be wrong. The user can use the model's credibility to adopt a coping strategy, which strengthens the resilience of the entire system in the face of attacks. Reducing environmental vulnerability from the environmental aspect improves access security, environmental confidentiality and privacy, and enables a better response to external attacks.
Explainable and interpretable	Explainability refers to a representation of the mechanisms underlying AI systems' operation, whereas interpretability refers to the meaning of AI systems' output in the context of their designed functional purposes. [b-NIST AI RMF 1.0]	Explainable and interpretable for AI systems can deepen human understanding of the decisions and decision-making processes of AI systems, so that security risks in AI systems can be better identified and addressed in system	–	Interpretability	–	Explainable and interpretable AI systems correspond to the model interpretability indicator

Table 1 – The security-related features in [b-NIST AI RMF 1.0] and corresponding AI evaluation indicators

Features	Definition	Degree of relevance to AI security	Data indicator	Model indicator	Environment indicator	Relevance of AI features and indicator
		design, management and auditing.				
Privacy enhanced	Privacy refers generally to the norms and practices that help to safeguard human autonomy, identity, and dignity. These norms and practices typically address freedom from intrusion, limiting observation, or individuals' agency to consent to disclosure or control of facets of their identities (e.g., body, data, reputation) [b-NIST AI RMF 1.0].	Attackers typically use sensitive data unintentionally leaked by AI systems as an entry point for attacks. Therefore, ensuring privacy can better protect the security of the AI system itself.	–	–	E_Completeness	Privacy requires data encryption and desensitisation from the environmental aspect to protect privacy, and the reduction of environmental vulnerabilities to avoid privacy breaches due to intrusion.
Fair	Fairness in AI includes concerns for equality and equity by addressing issues such as harmful bias and discrimination [b-NIST AI RMF 1.0].	By affecting the fairness of the AI system, an attacker can cause the AI system to behave in unexpected ways, thereby achieving the goal of the attack.	D_Diversity, D_Balance	M_Accuracy	–	Improving the fairness of AI systems requires improving the balance and diversity of data to ensure the fairness of model training, as well as evaluating the accuracy of models against different test sets to ensure the fairness of the system.

The proposed AI security evaluation framework in ICT is mainly carried out from three dimensions: data, model and environment.

- 1) Data security primarily ensures the security of the entire data collection, storage and processing process, with the following security indicators: completeness, accuracy, consistency, diversity and balance, as well as the comprehensive evaluation for data security;
- 2) Model security primarily ensures that the decision-making results of the model are reliable and controllable, with the following security indicators: accuracy, robustness, credibility and interpretability, as well as the comprehensive evaluation for model security;
- 3) Environment security primarily ensures the overall security of AI basic platform security and AI business service security. The security indicators are: vulnerability, completeness and robustness, as well as the comprehensive evaluation for environment security;
- 4) A comprehensive evaluation of the entire AI technology will be conducted by combining the evaluation results of the above three dimensions to provide a solid and reliable basis and reference direction for relevant decision-making and research on AI.

7 AI security evaluation methods

In this clause, the security of data, model and environment will be evaluated separately. In order to help readers understand the research content of this paper more conveniently and efficiently, Table 2 is given, which summarises the core points of this study in a systematic way, aiming to provide readers with a clear overview of the research lineage and key content.

Table 2 – The overview of the research lineage and key content

Aspect	Indicator	Item	Marking/Symbol
Data	D_Completeness	Null value detection	S_{D_Nul}
		Data mutation detection	S_{D_Fal}
	D_Accuracy	Anomaly detection	Statistics/Machine learning/Clustering algorithm/others++ S_{DA} calculation
	D_Consistency	Inconsistency in the format	S_{D_For}
		Content verification	S_{D_Con}
		Consistency of logical relationships	S_{D_Log}
	D_Diversity	Feature distance between samples	for sequence data and image data S_{DD1}
			for text dataset S_{DD2}
	D_Balance	For classification problems	Shannon index

Table 2 – The overview of the research lineage and key content

Aspect	Indicator	Item	Marking/Symbol
Model	M_Accuracy	For regression	Mean squared error+Root mean square error+Mean absolute error+R-squared score
		For classification	Confusion matrix Accuracy Precision Recall rate F1 score c_AUC
	M_Robustness	Adversarial classification accuracy	S_{M_ACA}
		Adversarial accuracy	S_{M_AA}
	M_Credibility	Credibility bias score for n -class-labelled data	S_{MC}
	M_Interpretability	Transparency score	The quantization table of interpretability
Environment	E_Vulnerability	Vulnerability evaluation score	The quantization table of vulnerability
	E_Completeness	Completeness evaluation score	The quantization table of completeness
	E_Robustness	Fault tolerance	S_{ER}

7.1 AI data security evaluation methods

In AI application scenarios, extracting valuable data from massive datasets has been a challenging problem in training effective AI models. This chapter analyses the influence of the entire lifecycle of training data on the security of AI technology. It identifies the indicators of data completeness, accuracy, consistency, diversity, and balance. Moreover, based on these security indicators, a comprehensive evaluation of the data is conducted to fundamentally enhance the performance of AI technology.

7.1.1 Completeness evaluation

- **Description:** Data completeness means that the data is not lost, damaged or tampered with during the processes of collection, storage, transmission and processing.
- **Necessity analysis:** Data completeness evaluation, on the one hand, can determine whether there is unauthorized tampering with AI training data. Such tampering can cause the model to learn the wrong data distribution, making it more vulnerable to false data injection attacks. On the other hand, it is to evaluate whether there are a large number of null values in the dataset. If there are numerous null values, the model will have difficulty learning accurate patterns and regularities, rendering it susceptible to adversarial sample attacks.
- **Example of necessity analysis:** Taking the AI model for Internet of things (IoT) device fault diagnosis as an example, the training data contains operating parameters such as temperature, voltage, and current of various IoT devices, as well as the corresponding device states (normal or faulty). Attackers launch a data injection attack and tamper with the training data, changing the status of multiple malfunctioning smart meters (with abnormal voltage, current fluctuations, etc.) from faulty to normal. When the model is trained with the tampered data,

it will learn the wrong relationship between the operating parameters and states of the device. In practical applications, when the operating parameters of a truly faulty IoT device or those after an elaborate attack by the attacker are input into the model, the model may misclassify the device as a normal device, thus failing to detect the device fault in time. When collecting the operating parameters of IoT devices as training data, due to sensor failures or communication interruptions, null values may appear in the temperature data of some devices. The model may focus too much on other non-null parameters and overlook the importance of temperature for device fault diagnosis. Attackers can take advantage of the weakness of the model's insufficient learning of temperature parameters to launch adversarial sample attacks.

- **Evaluation method:** Data completeness evaluation includes data mutation detection and null value detection.

Evaluation item1-data mutation detection: The injected false data usually appears random, while the real data typically conforms to a certain pattern of change. When false data is added to the original dataset, it will introduce a certain degree of variability to the data. Therefore, for data mutation detection, the rate of change of the root mean square of the data can be calculated. The degree of data variability determined in this way serves as one of the bases for judging whether the training data has been attacked by data injection.

$$S_{D-Fal} = \begin{cases} 0, & |c - c'| \geq \beta \\ (1 - \frac{|c - c'|}{c}) * 100\%, & |c - c'| < \beta \end{cases}$$

where c' denotes the root mean square of the current sample data, and c denotes the average of the root mean squares of multiple historical sample data of the same batch, β is the preset value. When the rate of change of the root mean square is too large, it indicates that there may be data mutations. It is worth noting that the generation of noise can also cause the rate of change of the root mean square to be too large, so further confirmation is needed when the data shows high mutability. The value of S_{D_Fal} is 0 for a dataset with large data variations and 100% for a dataset without data variations.

Item2-null value detection: Null value completeness focuses on the distribution of data with null values. This distribution can be considered as one of the important factors in evaluating the risk of the model being subjected to network attacks.

$$S_{D_Nul} = \frac{a - a'}{a} * 100\%,$$

where a' signifies the number of training data samples with null values and a indicates the size of data. The value of S_{D_Nul} is 0 for a dataset with many null-valued data and 100% for a dataset without null values.

Based on the results of null value detection and data mutation detection, data completeness S_{DC} can be given as:

$$S_{DC} = w_{d1}S_{D_Fal} + w_{d2}S_{D_Nul},$$

where w_{d1} and w_{d2} are the connection weights. The value of S_{DC} is 0 for a dataset with extremely poor data completeness and 100% for a dataset with perfectly good data completeness.

7.1.2 Accuracy evaluation

- **Description:** Data accuracy can measure the closeness of the collected data to the actual or expected value [b-Elouataoui].
- **Necessity analysis:** When there is anomalous data in the training data, the AI model may not be able to recognize the correct patterns, which will affect the accuracy of the model. The existence of anomalous data weakens the accuracy of the model's decision-making. Attackers

can take advantage of the model's sensitivity to anomalous data to generate adversarial samples, making the model more vulnerable to adversarial attacks. Evaluating data accuracy is to judge whether the model trained on this data is vulnerable to adversarial attacks.

- **Example of necessity analysis:** For example, in an AI model for automated driving image recognition, if the key features of images with signal-light signals are incorrect or mislabelled in the training data, the model will learn incorrect information for decision-making. Attackers can analyse the model's performance based on the inaccurate data and find that the model is more likely to make mistakes. Then, in a scene image where the traffic light should be green for passage, they can add physical interference to cause the camera to capture a slightly and elaborately designed pixel perturbation in the image. As a result, the model mistakes it for a red-light signal.
- **Evaluation method:** Anomaly detection is required to evaluate the accuracy of the data. There are many methods for data anomaly detection, and the following table lists some of the commonly used methods. Different methods have different advantages and disadvantages, and in practical scenarios, it is necessary to consider the characteristics of the data, the application scenario, the required accuracy, and computational resources to choose different detection methods.

Table 3 shows the methods for data anomaly detection.

Table 3 – The methods for data anomaly detection

Type	Detection method	Description
Statistics	3Sigma	Outliers are values that are more than 3 standard deviations away from the mean
	Z-score	After normalizing the data, those points whose Z-score is above a certain threshold in absolute value are outliers
	Boxplot	Outliers are identified by plotting five-digit generalizations of the data (minimum, first quartile, median, third quartile, maximum)
	Grubbs hypothesis testing	If the Z-score of a suspected point is greater than the Grubbs critical value, the suspected point is an outlier
	SOS	Stochastic outlier selection (SOS) identifies outliers by random sampling and statistical testing
Machine learning	K Nearest Neighbour	The distance between each point and its nearest neighbour is calculated, and then the threshold of the distance is used to determine whether it is an anomaly
	Local outlier factor	Outliers are identified by calculating the local density deviation at each point
	Isolation forster	Outliers are detected by building multiple isolation trees.
	One-class SVM	Support vector machines (SVM) are used to learn the normal pattern of the data, and then points that are significantly different from the normal pattern are identified as outliers.
	AutoEncoder	The autoencoder is used to learn the normal pattern of the data, and then the reconstruction error is calculated to identify the outliers.
Clustering algorithm	DBSCAN	Density-based spatial clustering of applications with noise (DBSCAN) clusters data points according to their density, and isolated points that are not contained in an effective clustering are treated as outliers.
Other	Rule-based approach	It defines the characteristics of outliers based on business rules or domain knowledge and then detects outliers through these rules.

After anomalous data is discovered by outlier detection technology, the accuracy of the training dataset can be given as [b-Elouataoui]:

$$S_{DA} = \frac{d - d'}{d} * 100\%,$$

where d' is the number of training data samples with outliers and d indicates the size of data. The value of S_{DA} is 0 for an extremely inaccurate dataset and 100% for a perfectly accurate dataset.

7.1.3 Consistency evaluation

- **Description:** Data consistency can be used to describe whether the same entity has the same attributes in different datasets, and whether each entity and attribute conforms to consistency constraints.
- **Necessity analysis:** For contradictory samples in training data, AI models cannot understand the correct relationships in the data, which reduces the predictive stability of the model in different scenarios and even at different times. Therefore, the consistency of the data needs to be evaluated. Data inconsistency can interfere with the model's feature extraction process and cause it to learn incorrect classifications, making the trained model more vulnerable to adversarial sample attacks and sample escape attacks.
- **Example of necessity analysis:** For example, in the training data of an intrusion detection model, there is one data record showing that source Internet protocol (IP) A communicates with destination IP B using protocol type transfer control protocol (TCP), port number 80, and the training data labels have contradictory normal and intrusion classifications. With such contradictory label information, the model cannot determine whether the pattern is normal or intrusive. During the testing process, the attacker observed and analysed the testing data, and found that the intrusion detection model was uncertain in judging the communication characteristics of source IP A and destination IP B on TCP 80 port, and the decision when processing specific hypertext transfer protocol (HTTP) response packets was still wrong. Therefore, the attacker can bypass intrusion detection by constructing malicious samples.
- **Evaluation method:** Data consistency evaluation is carried out by evaluating the proportion of inconsistent data in the dataset to the total amount of data, consisting of format consistency, content consistency and relationship consistency.

Item1-format consistency: Inconsistency in the format of the training data may lead to confusion in the AI model training process [b-Tian]. For example, in image recognition, inconsistent formatting of image sizes, colour channels, etc., may prevent the model from learning features correctly, which can make the model output unreliable and lead to accidents in safety-sensitive scenarios (e.g., recognition of traffic signs by autonomous driving). Thus, the format of each value in the dataset needs be verified:

$$S_{D_For} = (1 - \frac{e'}{e}) * 100\%,$$

where e' represents the number of instances that do not match the standard format and e indicates the size of data. The value of S_{D_For} is 0 for a dataset with extremely inconsistent data formats and 100% for a dataset with perfectly consistent data formats.

Item2-content consistency: When the content is inconsistent, such as when conflicting classification labels are assigned, it can cause the AI to generate false perceptions. Taking malware detection as an example, if the same software is labelled as normal and abnormal at the same time, it leads to confusion in AI decision-making and poses a huge threat to system security. Therefore, content verification should be evaluated, which involves verifying that the content data records open for the same matter are consistent and do not conflict [b-Tian].

$$S_{D_Con} = (1 - \frac{f'}{f}) * 100\%,$$

where f' represents the number of data records with conflicting content data and f indicates the size of data. The value of S_{D_Con} is 0 for a dataset with extremely poor consistency and 100% for a dataset with excellent consistency.

Item3-relationship consistency: The content of the data recorded in the dataset has a consistent relationship with the other content recorded, such as values, logical calculation relationships, etc. The consistency of logical relationships is equally important. In complex reasoning tasks, if the logical relationships of the data are inconsistent, the AI model may learn the wrong logic, affecting the accuracy and reliability of its decisions [b-Tian]. The consistency of relationships in data can be given as:

$$S_{D_Log} = (1 - \frac{g'}{g}) * 100\%,$$

where g' represents the number of data records with conflicting sample content data between datasets and g indicates the size of data. The value of S_{D_Log} is 0 for a dataset with extremely inconsistent data logic and 100% for a dataset with perfectly consistent data logic.

Therefore, based on the analysis, the data consistency S_{DCO} can be described as:

$$S_{DCO} = w_{d3}S_{D_For} + w_{d4}S_{D_Con} + w_{d5}S_{D_Log},$$

where w_{d3} , w_{d4} and w_{d5} represent the connection weights. The value of S_{DCO} is 0 for a dataset with extremely poor consistency and 100% for a dataset with excellent consistency.

7.1.4 Diversity evaluation

- **Description:** Diversity refers to a training dataset containing instances from various categories.
- **Necessity analysis:** Insufficient data diversity makes the model more vulnerable to adversarial samples. A training dataset lacking diversity may cause the model to be overly sensitive to specific patterns or features while having insufficient ability to recognize other patterns or features. When the model encounters adversarial samples, this flaw will make it difficult for the model to accurately distinguish the true nature of these adversarial samples [b-Li].
- **Example of necessity analysis:** For example, for AI technologies used to monitor potential security threats and fault points of IoT devices in real time, if the training data lacks diversity, the model will be able to accurately identify fewer types of attacks and fault patterns, further endangering the security and stability of IoT devices. Data diversity evaluation is one of the important considerations for evaluating whether a model may be subject to adversarial sample attacks and sample escape attacks.
- **Evaluation method:** A common method for data diversity evaluation is to analyse the feature distances between samples. The feature distance between samples can reflect the distribution of data in the feature space. The larger the feature distance, the more significant the differences between samples in the feature dimension. Therefore, data diversity can be measured by calculating the feature distances between samples in the dataset. Since sequence data and image data are mainly composed of numerical data, the Jaccard distance can be used to calculate the sample distances. As text data is predominantly non-numerical, the method of calculating n -gram overlaps is adopted to calculate the sample distances.

1) For sequence data and image data [b-Wu], the diversity index can be computed as:

$$S_{DD} = \frac{2}{h'(h'-1)} \sum_{h_i, h_j \in [C]} J(h_i, h_j) * 100\%, i \neq j,$$

where C is the dataset, $J(h_i, h_j)$ means the Jaccard distance of the i -th sample and the j -th sample and h' is the size of the dataset. The value of S_{DD} is 0 for an extremely nonlinearity dataset and 100% for a perfectly diverse dataset.

2) For text dataset [b-Reif], the diversity index can be computed as:

$$S_{DD} = \left(1 - \frac{2}{h'(h' - 1)} \sum_{h_i, h_j \in [C]} Sm(h_i, h_j)\right) * 100\%, i \neq j,$$

where C is the dataset, $Sm(h_i, h_j)$ measures the similarity between two instances h_i and h_j by computing the n -gram overlaps and dividing by the number of tokens in the longer text and h' is the size of the dataset. The value of S_{DD} is 0 for an extremely nonlinear dataset and 100% for a perfectly diverse dataset.

7.1.5 Balance evaluation

- **Description:** Data balance is the relatively average distribution of categories or features in a dataset.
- **Necessity analysis:** Data balance is an important characteristic of training data. If a discrete training dataset has significant differences in the number of each class, the classifier tends to favour the majority class and the prediction accuracy of the minority class is low; if a continuous training dataset has a highly skewed distribution, this also affects the accuracy of the data analysis, leading to discriminatory decision-making results from the AI model and affecting the model's ability to generalize, leading to poor model results on real scenario data. In addition to the natural data skewing in real-world scenarios and the data bias caused by wrong data sampling methods, attackers can also artificially control the probability distribution of the training data so that the model produces erroneous results, leading the model to be more vulnerable to adversarial attacks and sample escape attacks. This vulnerability is closely related to the model's learning pattern. The model tends to overlearn the features of the majority class and underlearn the features of the minority class.
- **Example of necessity analysis:** Taking the intrusion-detection AI model as an example, the model deeply learns the common patterns of normal traffic, such as the normal format of HTTP requests and the communication on common ports. However, for a new type of distributed denial-of-service (DDoS) attack, its traffic features, like the specific distribution of packet sizes and the distribution pattern of attack-source IPs, are not fully represented in the training data. Similarly, the training data lacks the traffic features of a certain new-type structured query language (SQL) injection attack, such as malicious SQL code. As a result, the model has difficulty accurately learning these features. Attackers may take advantage of this situation. For the SQL injection attack, they might insert a carefully crafted SQL-like injection statement into a normal HTTP request. By slightly adjusting some characters or formats, they make it resemble normal request parameters. Due to the model's overlearning of normal traffic and underlearning of SQL injection features, it may misclassify this adversarial sample as normal traffic. Regarding the DDoS attack traffic, attackers can manipulate the sending frequency and size of packets, making them approximate the pattern of normal traffic. Since the model's decision-making boundary is affected by data imbalance, it has a tendency to classify traffic as normal. Therefore, balance evaluation is one of the important considerations for evaluating whether the model may be subject to adversarial attacks and sample escape attacks.
- **Evaluation method:** The Shannon index is often used to evaluate the evenness of species in an ecosystem (the distribution of the number of individuals of each species). If the numbers of multiple species are relatively even, then the Shannon index will be relatively high, and vice versa.

The Shannon index [b-Shannon], [b-Vidakovic] can be computed as:

$$S_{DB} = - \sum (p_i * \log p_i) * 100\%,$$

where p_i represents the proportion of the i -th class in the total sample and $\log(\cdot)$ is the logarithmic function symbol. The value of S_{DB} is 0 for an extremely unbalanced dataset and 100% for a perfectly balanced dataset.

7.1.6 Data security comprehensive evaluation

Based on the analysis of completeness, accuracy, consistency, diversity and balance, the comprehensive evaluation of data security Ω_D is defined as:

$$\Omega_D = W_{D1}S_{DC} + W_{D2}S_{DA} + W_{D3}S_{DCO} + W_{D4}S_{DD} + W_{D5}S_{DB},$$

where W_{Di} ($i = 1, 2, 3, 4, 5$) is the connection weight and can be determined by expert knowledge and contribution degree. In fact, each security indicator is interdependent and interacts with the others, such as the completeness of data affecting the consistency of data, which in turn affects the accuracy of data. Therefore, it is necessary to comprehensively consider the weights of various evaluation indicators to make more scientific and reasonable decisions.

Analytic hierarchy process (AHP): In this manuscript, the analytic hierarchy process (AHP) will be introduced due to its simple principle and easy implementation.

Construction and verification of judgement matrix: In this AHP, for m evaluation indicators of data security, judgement matrix $\mathbf{C}$ is first constructed and the element c_{ij} of $\mathbf{C}$ are given using the 1-9 scale method proposed by Saaty [b-Saaty], using two-by-two comparison among indicators.

Table 4 illustrates the 1-9 scale method.

Table 4 – 1-9 scale method

Scale	Interpretation
1	Equally important compared to two dimensions
3	One dimension is slightly more important than the other
5	One dimension is clearly more important than the other
7	One dimension is more strongly important than the other
9	One dimension is more extremely important than the other
2, 4, 6, 8	Median of two adjacent dimensions
Inverse	$c_{ji} = 1 / c_{ij}$

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \dots & c_{1m} \\ c_{21} & c_{22} & \dots & c_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mm} \end{bmatrix},$$

Then, a consistency testing is performed for matrix $\mathbf{C}$:

$$CR = \frac{CI}{RI},$$

$$CI = \frac{(\lambda_{\max} - \tau)}{(\tau - 1)},$$

where CR is the consistency ratio, $\lambda_{\max}$ is the absolute value of the largest eigenvalue of the matrix $\mathbf{C}$ and τ is the number of non-zero eigenvalues in the order, RI is the random consistency index. It can be directly obtained from Saaty and summarized in Table 5. As shown in Table 5, when the order of the matrix is 4, the value of RI is 0.9. If $CR < 0.1$, the matrix passes the consistency verification, otherwise it does not have satisfactory consistency, and the judgement matrix $\mathbf{C}$ needs to be adjusted.

Table 5 – The value condition of RI

Order of matrix	3	4	5	6	7	8	9	10
<i>RI</i>	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

Normalization of weight vector: The judgement matrix **A** passes *CR* verification, then the eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$ is the requested weight vector, and the corresponding weight of each index is derived through the normalization operation.

Fuzzy analytic hierarchy process (FAHP): The assignment of quantitative values for the element c_{ij} in the judgement matrix **C** may be subjective and uncertain, because it relies on the expert's individual experience and knowledge. Different experts may arrive at different weights or even conflict. Especially when there are many factors (more than 6), the workload of scaling is too large, and the process cycle is long and complex, which is more likely to cause confusion in judgement, reduce the quality of the results of the AHP method, and even lead to the failure of AHP method decision-making. In fact, due to the inherent vagueness and subjectivity of some dimension values, it is difficult to evaluate multidimensional problems.

Based on a fuzzy affiliation function, the fuzzy factor transforms qualitative evaluation into quantitative evaluation and solves the problem of difficult quantification and uncertainty by harmonizing and unifying the subjective and objective criteria, so as to evaluate things or objects subject to multidimensional constraints realistically. After the introduction of the fuzzy factor, the element a_{ij} of fuzzy judgement matrix **A** is:

$$a_{ij} = \sum x_{ij}F(x_{ij}), \quad \mathbf{A} = \begin{bmatrix} a_{11}, & a_{12}, & \dots, & a_{1m} \\ a_{21}, & a_{22}, & \dots, & a_{2m} \\ & & \mathbf{O} & \\ a_{m1}, & a_{m2}, & \dots, & a_{mm} \end{bmatrix}$$

where $F(\bullet)$ is the fuzzy affiliation function and can be selected according to the specific scenario with a suitable affiliation function, such as an expert voting decision, Gaussian function, gradient function, etc., x_{ij} is the quantized value for the comparison of indicator i and indicator j , and the final quantized value is the sum of the product of the raw quantized value and the membership degree. For example, if the equalization and accuracy are equally important, after the calculation of the affiliation function, the degree of affiliation is 0.5, and the quantitative score is that $1 \times 0.5 = 0.5$. If equalization and accuracy are slightly important, the degree of affiliation of the equal importance of balance and accuracy is also 0.5, the quantitative score is that $3 \times 0.5 = 1.5$. Then, the final quantitative score is that $0.5 + 1.5 = 2$. Therefore, the introduction of the fuzzy factor in Saaty 1-9 proportionality scale can be simplified as shown in Table 6.

Table 6 – Scale of proportions of F-AHP

Indicator i compares indicator j	Quantitative value
Equal importance	1
Slightly important	3
More important	5
High priority	7
Vital	9

The introduction of fuzzy theory can minimize and mitigate the errors caused by subjective judgements on the evaluation results, but these errors cannot be avoided completely. Therefore, the consistency ratio (CR) is still calculated on the judgement matrix **A** to further ensure the dependability of the decision. Upon passing the CR verification, the eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$ is determined as the weight vector, and the relative weights of each index are normalized to derive the final weights.

7.2 AI model security evaluation methods

For AI models, accuracy serves as a genuine indicator of the model's performance and forms the foundation for their practical applicability in real-world production. Robustness refers to the stability of a model when confronted with noise, interference, abnormal data, and other challenging scenarios, which is crucial for the implementation of AI in practical applications. The extensive utilization of AI has prompted apprehensions regarding the existence of vulnerabilities in the model, the potential for hijacking and manipulation of parameters, and the risk of key parameters being leaked. The calibration of confidence (referring to the cognitive certainty of the model) and interpretability are also crucial factors in addressing uncertainties regarding the decision-making outcomes of AI. The importance of security in AI models is on the rise. Therefore, this proposal evaluates the security of AI models based on model accuracy, robustness, credibility and interpretability.

7.2.1 Accuracy evaluation

- **Description:** Accuracy refers to the closeness between model prediction or classification results and actual observations.
- **Necessity analysis:** Low-accuracy models are more vulnerable to adversarial sample attacks. Since low-accuracy models do not learn the classification boundaries well and there are many misclassified areas, attackers can analyse and create adversarial samples that cause the model to make wrong classifications. Low-accuracy models cannot accurately capture the key features of malicious samples.
- **Example of necessity analysis:** For malicious samples with ambiguous features, whose features overlap partially with normal samples or whose malicious features are extremely obscure, the model has difficulty accurately extracting and identifying them from a complex set of features. Therefore, the model is more vulnerable to sample escape attacks. Evaluating the accuracy of the model can reveal the potential vulnerabilities of the model when facing adversarial sample attacks or malicious sample escape attacks. The example description in clause 7.1.5 also applies to this clause. Model accuracy evaluation is completed by evaluating the accuracy of the model's judgement on the testing data.
- **Evaluation method:** Different AI models may need to use different evaluation indexes, and appropriate evaluation indexes should be selected according to specific application scenarios and model types. At the same time, the evaluation indicators should also be selected with the characteristics of the actual problem and dataset to ensure the reliability and validity of the evaluation results. Some commonly used evaluation indicators are summarized in Table 7.

Table 7 – The commonly used evaluation items for accuracy

Item name	Definition	Application	Description of value
Mean squared error	$\Sigma(\text{value of predicted-value of true})^2 / \text{number of samples}$	Regression	$0 \sim +\infty$, lower values are better
Root mean square error	The square root of the mean squared error	Regression	$0 \sim +\infty$, lower values are better
Mean absolute error	$\Sigma \text{value of predicted-value of true} / \text{number of samples}$	Regression	$0 \sim +\infty$, lower values are better
R-squared score	$1 - \Sigma(\text{value of predicted-value of true})^2$	Regression	$-\infty \sim 100\%$, high values are better
Confusion matrix	True positive (TP)	Classification	0% ~ 100%, high values are better
	True negative (TN)		0% ~ 100%, high values are better
	False positive (FP)		0% ~ 100%, lower values are better
	False negative (FN)		0% ~ 100%, lower values are better

Table 7 – The commonly used evaluation items for accuracy

Item name	Definition	Application	Description of value
Accuracy	$((TP+TN) / \text{Total samples}) \times 100\%$	Classification	0% ~ 100%, high values are better
Precision	$(TP) / (TP+FP) \times 100\%$	Classification	0% ~ 100%, high values are better
Recall rate	$(TP) / (TP+FN) \times 100\%$	Classification	0% ~ 100%, high values are better
F1 score	$2 * \text{Precision} * \text{Recall rate} / (\text{Precision} + \text{Recall rate}) \times 100\%$	Classification	0% ~ 100%, high values are better
e_AUC	The area under the receiver operating characteristic (ROC) curve	Classification	50% ~ 100%, closer to 100% indicates better classifier performance

In order to evaluate the accuracy of the model more comprehensively, for the same application scenario, more than one item may be used, so the accuracy score of the model can be given as:

$$S_{MA} = \frac{1}{(m+n)} \left(\sum_{i=1}^m (1-r_i) + \sum_{j=1}^n r_j \right) * 100\%,$$

where r_i denotes the value of the i -th item that is better with higher performance and m indicates the number of item, r_j denotes the value of j -th item that is better with lower performance and n indicates the number of item.

7.2.2 Robustness evaluation

- **Description:** Robustness refers to the ability of a model to maintain stable performance (such as accuracy) in the face of minor changes in input data, noise interference, or other potential disturbing factors.
- **Necessity analysis:** For models with low robustness, their decision boundaries are often relatively sensitive. Such models are highly susceptible to noise attacks. Since their internal feature extraction and decision-making mechanisms are prone to being disrupted when there is noise in the samples, the models make incorrect decisions. Attackers can cause the model to misjudge simply by adding noise to the input data of the model without having to deeply understand the structure and parameters of the model.

Moreover, these models are also vulnerable to adversarial sample attacks. Attackers can use specific algorithms to explore input samples near the decision boundary and determine the direction of small perturbations required for these samples to cross the decision boundary, thus causing the model to misclassify.

- **Example of necessity analysis:** Taking the object-recognition AI model in an autonomous driving system as an example. During the image data transmission process, it is inevitably affected by noise such as electromagnetic interference. If the model has poor robustness, even a small amount of random noise in the image may cause the model to misjudge the object during the object-recognition process. For instance, it may misclassify a pedestrian as another object or fail to recognize the object at all, thus making incorrect decisions. The example in clause 7.1.2 also applies to explaining the adversarial sample attacks mentioned in this clause. To evaluate the robustness of a model, adversarial samples and noisy samples are usually used to analyse the decision-making accuracy of the model.
- **Evaluation method:** The adversarial classification accuracy and adversarial accuracy can be used to evaluate the robustness of an AI model from different concerns.

- 1) Adversarial classification accuracy is primarily used to measure the ability of an AI model to correctly categorize carefully designed adversarial samples. If a model has a high classification accuracy rate for adversarial samples, it indicates that it has a better resistance to adversarial attacks, can recognize and handle these samples with disturbing properties to a certain extent, and will not be easily misled to produce incorrect classification results, reflecting the stability of the model in the face of malicious attacks or data perturbation. The adversarial classification accuracy can be calculated as the percentage of correctly categorized adversarial samples out of the total number of adversarial samples [b-Sun].

$$S_{M_ACA} = \frac{u'}{u},$$

u' is the number of correctly classified adversarial samples, and u is the number of all adversarial samples. The value of S_{M_ACA} is 0 for a poorly robust model and 100% for a more robust model, representing the ability of the model to cope with adversarial samples or noise.

- 2) Adversarial accuracy is used to evaluate the attack success rate of the adversarial samples. It mainly reflects the probability of success of different types of adversarial sample attacks on the model and expresses the vulnerability of the model to different attacks. Higher adversarial accuracy means that the model is vulnerable to specific attacks with specific types of vulnerabilities, which may be exploited maliciously in practical applications, leading to problems such as wrong decisions or information leakage. This accuracy can be further subdivided into adversarial accuracy for white box attacks (AAW), and adversarial accuracy for black box attacks (AAB), depending on the type of attack. The adversarial classification accuracy can be calculated as the percentage of total adversarial samples that were misclassified [b-Zituo].

$$S_{M_AA} = \frac{v'}{v},$$

where v' is the number of adversarial samples for which the prediction was incorrect, and v is the number of all adversarial samples. The value of S_{M_AA} is 0 for a more robust model and 100% for a poorly robust model. It is worth noting that S_{M_ACA} plus S_{M_AA} equals 1, thus different evaluation indicators should be adopted according to different emphases.

7.2.3 Credibility evaluation

- **Description:** Credibility reflects the degree of trust in the output of an AI model, and refers to the cognitive certainty of the model, i.e, the degree to which the model's output of the maximum distribution probability value for each class through the softmax function matches the model's actual probability of correct prediction.
- **Necessity analysis:** When a model has low credibility, it indicates that the model is less certain about its judgements for most input samples, making it more vulnerable to adversarial sample attacks. Attackers can search for minor changes around the input samples, which may cause significant fluctuations in the model's credibility, thus leading the model to make wrong decisions. Evaluating a model's credibility can help determine whether the model is prone to adversarial sample attacks.
- **Example of necessity analysis:** Taking the intrusion-detection AI model as an example, for some new types of network scanning behaviours, these scans exhibit features similar to those of normal network maintenance tools, as well as signs resembling common intrusion scans. When processing such traffic samples, the probability that the model outputs a judgement of malicious traffic fluctuates within an unstable range, such as between 0.3 and 0.7, and the model cannot clearly determine whether the traffic is normal or intrusive. Specifically,

attackers can disguise malicious DDoS attack traffic. Within the framework of normal network communication protocols, they fine-tune the packet sending frequency, the size distribution of packets, and the values of some protocol header fields. For instance, they adjust the sending frequency of DDoS attack packets to be close to the request frequency of normal network applications under high-concurrency conditions. Meanwhile, they modify certain identifiers in the packets to make them appear more like normal business data requests. After such elaborate disguise by attackers, the probability that the model judges this traffic as malicious drops from 0.6 to around 0.2, and thus the model determines it to be normal traffic.

- **Evaluation method:** The model's credibility can be measured by the degree of difference between the model's predicted class probabilities and the actual classification results.

For n -class labelled data [b-Guo], the formula for the credibility bias score is:

$$S_{MC} = \frac{1}{n} \sum_{i=1}^n (1 - |\delta_i - \delta'_i|) * 100\%,$$

where δ_i denotes the true classification accuracy of the model in the i -th class, and δ'_i is the average of the maximum of the probability distribution of the softmax output in the i -th class. n is the sample size of the i -th class. When S_{MC} is closer to 100%, it means that the output uncertainty of the AI model is smaller, the credibility is higher, and the output result is more reliable.

7.2.4 Interpretability evaluation

- **Description:** Interpretability refers to the ability to understand an AI model's decision-making process, including the reasons for making the decision, the methodology and the content of its decision.
- **Necessity analysis:** A model lacking interpretability is more vulnerable to black-box attacks. For such a model, security personnel face it like a "black box". It is difficult for them to identify all the samples that lead the model to make wrong decisions, giving attackers the opportunity to find input categories or patterns that can easily make the model err. In contrast, when a model has good interpretability, its coefficients and decision rules are clear, enabling security personnel to take corresponding protective measures.
- **Example of necessity analysis:** Taking the intrusion-detection AI model as an example, when dealing with traffic containing a specific encryption protocol, the model may easily misclassify malicious encrypted tunnel traffic (used to transfer stolen data) as normal VPN connection traffic. Since security personnel cannot deeply understand the model's judgement basis for encryption-protocol-related features, it is difficult for them to optimize the model's decision-making process. Attackers can then exploit this loophole by continuously sending malicious encrypted tunnel traffic disguised as normal VPN traffic, successfully bypassing the intrusion-detection system.
- **Evaluation method:** Interpretability evaluation involves quantitatively scoring the interpretability of the model's decision-making process. There are many ways to achieve interpretability [b-Paul], such as using gradient heatmaps, class activation maps and other methods for interpretable image classification to help users understand what image regions the AI model focuses on in the decision-making process; for the language generation class of the AI model, word frequency statistics and keyword extraction can be used to calculate the importance of each keyword in the generated text to help users understand the decision-making process of the model in generating text. An attention mechanism can also be used to visualize the model's attentional weights in the text generation process to show which input information the model is focusing on when generating each word; in addition, feature importance analyses can be used [b-Han], such as LIME evaluations (local interpretable

model-agnostic explanations), DeepLIFT evaluation (deep learning important features), Shap evaluation (Shapley additive explanations) methods, etc., to calculate the contribution of features to the prediction results, and thus to explain the extent of the model's dependence on different features. Further, the decision-making process of a model can also be explained by converting the model into a set of decision rules, such as a decision tree that demonstrates the decision-making process through logical and clear if-then-else statements, and a linear model such as linear regression that is easy to understand due to its concise input-output relationships. In addition, interviews and questionnaires with experts and scholars in the field of AI, enterprises, and users are used to collect their feedback on the interpretability of the models. Through the above methods, the degree of transparency of the model S_{MT} is explored so as to evaluate the interpretability of the AI model.

Table 8 illustrates the quantization table of interpretability.

Table 8 – The quantization table of interpretability

Score	Requirements for competencies
10	AI models do not support any interpretable methods
40	AI models can be explained in small parts by interpretable methods
60	AI models can be partially explained by interpretable methods
80	AI models can be mostly explained by interpretable methods
100	AI model can be fully explained by interpretable methods
25, 50, 70, 90	Intermediate between the above judgements

For instance, in the case of malware identification, the method described in clause 7.2.5 can be utilized to explain the model's recognition results. For example, through feature importance analysis, it is possible to identify which features (such as file size, file type, the occurrence of specific strings, etc.) play a crucial role in distinguishing between malware and benign software. Then, local explanation methods (such as LIME or SHAP) can be used to provide detailed explanations for individual samples, clarifying why the model identified them as malware or benign software. Additionally, global explanation methods and visualization techniques can be employed to further explain the model's decision-making logic and overall performance. The interpretability of AI models is of great significance in malware identification. By adopting feature importance analysis, local explanation methods, global explanation methods, and visualization techniques, clear explanations and decision-making bases can be provided to users, thereby enhancing the model's trustworthiness and accuracy.

7.2.5 Model security comprehensive evaluation

The comprehensive evaluation methods of models with multiple indicators can be found in clause 7.1.6.

7.3 AI environment security evaluation methods

When artificial intelligence models are manipulated and attacked, they have the potential to cause a cascading effect of immeasurable harm. Hence, it is essential to prioritize the security of the deployment environment of AI systems. This proposal evaluates the security of the AI environment by examining three key security elements: vulnerability, completeness, and robustness.

7.3.1 Vulnerability evaluation

- **Description:** Environmental vulnerability focuses on the risk evaluation of vulnerabilities faced during the development, deployment, and application of AI systems, covering hardware, software, operating system, and code.

- **Necessity analysis:** AI systems involve complex data flows and algorithms, which pose risks such as data leakage, model poisoning, and adversarial attacks. Vulnerability evaluation can help identify potential security vulnerabilities and threats in AI systems, enabling the deployment of preventive strategies in advance to safeguard against data breaches and system attacks.
- **Example of necessity analysis:** Versions of TensorFlow prior to 1.15.4, 2.0.3, 2.1.2, 2.2.1, and 2.3.1 contain a security vulnerability. This vulnerability arises from insufficient validation of the `data_splits` parameter in `tf.raw_ops.StringNGrams`. Attackers can exploit this vulnerability to cause memory data leakage. When the `StringNGrams` function is invoked, attackers can read sensitive data (e.g., return addresses) from memory by constructing specially crafted inputs. Attackers can directly exploit model interfaces to perform data poisoning, parameter theft, malicious file uploads, and deletion of critical components. These actions compromise the core data, algorithmic integrity, and operational stability of model services.
- **Evaluation method:** Vulnerability evaluation for hardware, software, operating system, and code is conducted. Based on [b-GB/T 43782-2024], the scoring items are summarized in Table 9, where items can be added, deleted and measured based on the real-world evaluation scenarios. It is worth noting that the sum of all items should equal 100%.

Table 9 – Scoring table of vulnerability evaluation

Category	Evaluation item	Description	Score (%)
Hardware	Debugging interface security	Strict debugging interface security mechanisms have been established and implemented.	s_1
	Chip security	Strict chip security mechanisms have been established and implemented.	s_2
	Trusted computing	Trusted computing has been implemented.	s_3
Software	Vulnerability patching	Ensure that there are no high-risk vulnerabilities related to the software published two months ago.	s_4
		Software exception monitoring has been implemented.	s_5
Operating system	Vulnerability patching	An automatic and mandatory upgrade mechanism has been established to eliminate security vulnerabilities.	s_6
		Ensure that there are no high-risk vulnerabilities related to the system published two months ago.	s_7
		Effective system exception monitoring has been implemented to promptly identify issues.	s_8
	Permission restriction	The principle of least privilege is adhered to. Users only have the minimum permissions required to perform their tasks, and all unauthorized permissions are prohibited.	s_9
	Boot security	System secure boot authentication and validation strategies have been set.	s_{10}
		System secure boot authentication and validation strategies are effective.	s_{11}
	Update security	The system supports both manual and automatic updates.	s_{12}
		Before upgrading, version checking, MD5 checking, and signature checking of the image are performed.	s_{13}
		When system updates are turned off or fail, it can roll back to the pre-update version.	s_{14}

Table 9 – Scoring table of vulnerability evaluation

Category	Evaluation item	Description	Score (%)
	Configuration security	System partition configuration security measures have been implemented.	S_{15}
		Default closing of debugging ports, default closing of remote login ports, and default closing of debugging log information services.	S_{16}
		Reasonable security attributes have been established and configured for input and output ports.	S_{17}
		Access control mechanisms have been implemented to prevent unauthorized calls to system interfaces.	S_{18}
		Login passwords meet complexity requirements.	S_{19}
Code	Code audit	Key codes have passed security audits.	S_{20}

7.3.2 Completeness evaluation

- **Description:** It is mainly used to evaluate the environmental completeness issues faced by AI systems during development, deployment, and application, and it focuses on ensuring that key components such as data, source programs, model files, configuration files, and dependency libraries have not been accidentally modified or maliciously tampered with during the deployment process as well as during subsequent operation.
- **Necessity analysis:** Completeness prevents environments from being maliciously compromised, unauthorizedly modified, or accidentally deviated, thereby avoiding AI system failures or erroneous outputs. The completeness evaluation of the environmental prevents the AI system environment from being maliciously compromised, from undergoing unauthorized modifications, or from experiencing accidental deviations that can cause the AI system to fail or output incorrect results.
- **Example of necessity analysis:** Before deploying models, an autonomous driving company must verify the firmware signatures of in-vehicle AI chips, hash values of container images, and logical integrity of sensor input validation. During runtime, model computations are protected via a trusted execution environment (TEE), with real-time monitoring for malicious code injection in memory-any integrity lapse in these steps may trigger accidents.
- **Evaluation method:** It can be carried out by evaluating the protection of application data and communication processes to detect whether they have been accidentally modified or maliciously tampered with. Referring to [b-ISO/IEC 42001], the evaluation items are summarized in Table 10. The items can be added, deleted and measured based on the real-world evaluation scenarios. Meanwhile, the sum of all items should equal 100%.

Table 10 – Scoring table of completeness evaluation

Category	Evaluation item	Description	Score (%)
Application data	Secure encryption	Important data in logs is encrypted.	x_1
		Sensitive data such as database connection passwords, file transfer protocol (FTP) service passwords, login passwords, and external system interface authentication passwords stored in context files (including configuration files, databases, cookies, etc.) during the application process are encrypted.	x_2
	Secure authentication	Resistant to brute-force attacks.	x_3
		Security checks and risk evaluations have been implemented.	x_4
	Access check	Strict access-role checking mechanisms have been implemented.	x_5
	Security protection	It has the ability to resist malicious tampering and malicious analysis.	x_6
	Transmission security	Data transmission security measures have been implemented to ensure the confidentiality, integrity, and validity of data.	x_7
	Storage security	Important data is stored encrypted.	x_8
		Important data is backed up, and the backup mechanism meets security requirements.	x_9
		Important data is stored in dedicated data hosts or storage media.	x_{10}
		Between device identification information, device configuration data, device parameter information, operating system data, and application software data, at least logical isolation from other categories of data has been implemented.	x_{11}
	Access control security	Unauthorized entity data access can be terminated, and perceptible alarm functions are provided.	x_{12}
		Access to or attempts to modify data are knowable and controllable.	x_{13}
Communication process	HTTPS filtering security	Blacklist filtering and interception of hypertext transfer protocol secure (HTTPS) have been implemented.	x_{14}
		Strategy cache management has been implemented.	x_{15}
	IP filtering security	IP blacklists and whitelists have been set. IP source or destination addresses in IP data packets forwarded by routers are filtered.	x_{16}
		Support for filtering of transfer control protocol (TCP), user diagram protocol (UDP), and Internet control message protocol (ICMP) protocols.	x_{17}
	Blacklist and whitelist	Cloud-based blacklist and whitelist management has been implemented.	x_{18}

Table 10 – Scoring table of completeness evaluation

Category	Evaluation item	Description	Score (%)
	management security		
	Cloud database malicious interception	Cloud-based HTTPS query and interception of malicious databases have been implemented.	x_{19}
	Trusted boot and static measurement	Trusted boot functions have been implemented.	x_{20}
		File static measurement has been implemented, including ELF files, SO files, kernel module KO files, SH files, and config files.	x_{21}

7.3.3 Robustness evaluation

- **Description:** Environment robustness refers to the fault tolerance of the AI model deployment environment in response to various types of failures such as hardware failures, network outages, and software errors.
- **Necessity analysis:** The evaluation of environment robustness ensures that the AI system can still run stably when a single point of failure, such as server downtime, network outage, dependency library conflict, operating system error, third-party service crash, etc., does not affect the overall service.
- **Example of necessity analysis:** Autonomous driving systems are highly dependent on multiple sensors to collaboratively perceive the environment, and any temporary failure of key sensors that is not dealt with in a timely manner by the system's fault-tolerance mechanism may trigger a chain of fatal accidents.
- **Evaluation method:** It can be tested under different failure scenarios, such as simulated hardware failures, network outages, etc., and the robustness can be evaluated by counting the AI system's task completion rate under these scenarios:

$$S_{ER} = \frac{z'}{z} * 100\%,$$

where z' is the number of times the AI system successfully completed the task and z is the total number of tasks, under the preset fault conditions. The higher the score of S_{ER} , the better the robustness of the system.

7.3.4 Environment security comprehensive evaluation

The evaluation methods with multiple indicators can be found in clause 7.1.6.

7.4 Comprehensive evaluation methods

Comprehensive evaluation of data, model and environment have been given respectively. Additionally, a comprehensive evaluation of the entire AI technology will be conducted by combining the evaluation results of the above three dimensions to provide a solid and reliable basis and reference direction for relevant decision-making and research on AI. The overall security Θ_{sys} of an AI system can be defined as:

$$\Theta_{sys} = W_D \Omega_D + W_M \Omega_M + W_E \Omega_E,$$

where W_D , W_M and W_E denote the weights of data, model, and environment, respectively, and Ω_D , Ω_M , and Ω_E are the security of data, model, and environment.

7.5 Conclusion

In this Technical Report, a systematic AI security evaluation framework is proposed with the necessity analysis, covering the data, models, and environmental indicators required for security, as well as the comprehensive evaluation methods for AI systems. The developers, users, and evaluation audit institutions of AI systems can use the specialized evaluation methods provided in clause 7.1, clause 7.2, and clause 7.3 respectively to achieve precise evaluation of the data security, model security and environmental security of an AI system; they can also use the comprehensive evaluation method proposed in clause 7.4 to measure the overall security level of AI systems, considering their decisions on the security indicators based on factors like user age or region. In addition, this report provides a detailed use case of the entire process from evaluation preparation to implementation through complete examples in Annex A, helping readers quickly grasp the key points of the application of this systematic AI security evaluation framework.

Annex A

Application use cases of AI security evaluation methods

At present, the application of machine learning techniques, especially deep learning algorithms, in the field of intrusion detection has gradually become a research hotspot. Machine learning technology can automatically extract potential abnormal behavioural features and patterns through the deep mining and learning of massive network data, thus realising the effective detection and early warning of unknown network intrusion behaviours and providing more intelligent and efficient solutions for network security protection.

Although the network intrusion detection methods based on machine learning have improved detection accuracy and efficiency to a certain extent, there are still some challenges and problems. First, in the data dimension, the effective training of intrusion detection models cannot be separated from a large amount of network tagged data, which may be subject to various interferences and damages in the process of data storage, transmission and use, such as data tampering, poisoning, deletion, or damage, resulting in serious deviations from the model's prediction and decision-making, which can lead to major network security accidents. Secondly, the security of the AI system's own algorithm has become a hot issue in current research. For example, in practical application scenarios, attackers may launch counterattacks against AI models by cleverly adding tiny but well-designed perturbations to the input data, thus inducing the AI system to produce erroneous output results, thereby successfully bypassing the intrusion detection mechanism and creating a huge hidden danger for network security. In addition, in the dimension of deployment environment, the operation of AI systems usually relies on certain physical facilities, such as servers, data centres and so on. If these physical facilities are attacked or damaged, it will lead to the paralysis of AI systems and cause serious problems such as data leakage, which will undoubtedly bring great economic losses and reputational damage to related organizations and individuals. Therefore, to address the above issues, this report will start from the three key dimensions of data, model and environment, respectively, and conduct an in-depth security evaluation, so as to ensure the reliability and stability of AI technology in the practical application of network intrusion detection.

A.1 Data security evaluation for intrusion detection

This security evaluation focuses on an intrusion detection system (IDS) developed and trained by a company and built on artificial intelligence technology. The distribution of relevant data has been integrated and summarized in Table A.1. As can be seen from the table, there are 72 input features, including various aspects of network traffic, such as the basic information of the network connection (source IP, destination IP, source port, destination port, etc.), the timestamps of the packets, the protocol type, the size of the traffic, etc. The analysis of these features can reflect the basic attributes of the network connection, which can be analysed through the analysis of these features. These features can reflect the basic attributes of network connections, and abnormal network behaviours can be detected by analysing these attributes. In addition, its labelled data covers seven major attack types in various intrusion detection scenarios such as DOS attacks, DDOS attacks and Web attacks. In terms of data volume, the training data totalled 5 102 978 items and the testing data 966 229 items. Based on the actual situation of these data, a comprehensive evaluation of the system's data security will be carried out in-depth, providing solid data support and a decision-making basis for the optimization and improvement of the system.

Table A.1 shows the distribution of data in IDS.

Table A.1 – Distribution of data in IDS

Data	Feature	Total number	Normal	DOS	DDOS	Web attack	Infiltration	Malware attack	Brute force attack	PortScan
Training data	72	5 102 978	3 880 605	684 539	235 639	1 542	202	16 020	30 559	253 872
Testing data	72	966 229	232 800	410 725	141 384	925	122	9 612	18 336	152 325

A.1.1 Determination of evaluation indicators of data security for intrusion detection

IDS detects intrusion attempts by monitoring and analysing data from networks and systems. However, in large-scale heterogeneous and complex networks, data completeness faces significant challenges. For example, when an intrusion occurs, hackers often take advantage of the large amount of useless data traffic and adopt DDOS attacks in an attempt to put the IDS into an overloaded operating state, causing significant delays or even complete interruption of data collection, making it difficult to obtain and update critical information in the network in a timely manner and bringing great challenges to the security analysis of the AI analysis module in the subsequent IDS. The accuracy of the data has a direct impact on the decision-making results of the intrusion detection model. In addition, the consistency, diversity and balance of data are quite important for AI models. Intrusion detection systems may collect data from multiple sensors, log sources, etc. Data from different sources may differ in format, content, etc. Therefore, data conflicts are avoided through consistency evaluation to keep the data at a uniform standard in different storage and processing sessions. Diversity, on the other hand, is reflected in the richness of the data, covering various types of network data and attack scenarios, which helps to improve the adaptability of the intrusion detection model. Balance ensures that the data is evenly distributed across different categories and features, preventing the detection model from being biased towards a particular domain. Next, the evaluation of each indicator is explained in detail.

A.1.2 The security indicators evaluation of data security for intrusion detection

The evaluation results of the intrusion detection data for each security indicator are summarized in Table A.2. A more detailed description of each of the evaluated indicators follows.

Table A.2 – The evaluation results of security indicators for intrusion detection data

Indicator	Completeness	Accuracy	Consistency	Diversity	Balance
Score	95.06%	99.92%	99.95%	80.65%	82.35%

Completeness: Firstly, null value detection is carried out; there are 2355 samples containing NaN in the whole dataset, so the null value detection score is $(5\,102\,978 - 2355)/5\,102\,978 \times 100\% = 99.95\%$. Then, after removing the samples containing a null value, data mutation detection, by calculating the root mean square of the current sample 621.46 and calculating the average root mean square value of 10 historical samples of the same batch is 689.3, so the data mutation detection score is $(689.3 - 621.46)/689.3 \times 100\% = 90.16\%$. In this completeness, it is stipulated that all completeness indicators have the same weighting for each completeness item, so the overall score for Completeness is $0.5 \times 99.95\% + 0.5 \times 90.16\% = 95.06\%$.

Accuracy: The dataset for intrusion detection is large in size, so the simple and easy-to-administer 3Sigma principle is used to detect outliers and thus evaluate the accuracy of the data. A total of 4026 samples were found to contain outliers, so the accuracy score was $(5\,102\,978 - 4026)/5\,102\,978 \times 100\% = 99.92\%$.

Consistency: The consistency of format was detected first, and 0 items were found, so the score of this item is $(1 - 0) \times 100\% = 100\%$. In the consistency of content detection, 216 items were found to have label inconsistency, so the score was $(1 - 216/5102978) \times 100\% = 99.99\%$. In addition, in the consistency of relationship detection, 6268 logical relationships were found to have abnormal data, so the score for this item was $(1 - 6268/5102978) \times 100\% = 99.87\%$. The overall score of consistency is $0.33 \times 100\% + 0.33 \times 99.99\% + 0.33 \times 99.87\% = 99.95\%$.

Diversity: In this manuscript, Jaccard distance is used to calculate diversity. Assuming there are 2 datasets A[1 1 2 5 8 1] and B[-1 1 2 6 7 0], the Jaccard distance between A and B is $1 - (2/8) = 0.75$, where 2 is the number of elements in the intersection of A and B and 8 is the number of elements in the concatenation of A and B. Using the same method, the intrusion detection dataset has 73 elements per sample, including 72 input features plus labelled values, traversed $5102978 \times (5102978 - 1)/2 = 13020189682753$ times, and the result is 80.65% after the results are summed and averaged.

Balance: In this intrusion detection dataset of this evaluation, there are 8 types of labels (1 normal + 7 attack types), and the distribution probabilities of each type are calculated separately, which are, in order, 0.76, 0.13, 0.05, 0.0003, 0.00004, 0.003, 0.006, 0.497. Then, Shannon entropy calculation is performed: 0.208, 0.265, 0.1498, 0.0024, 0.0004, 0.0174, 0.0307, 0.1498, and the result of the summation is 0.8325. Therefore, the score of balance is 82.35%.

A.1.3 Comprehensive evaluation of data security for intrusion detection

Based on the evaluation scores of the above security indicators, the comprehensive evaluation of data security is performed next.

Method 1: If all the security completeness indicators are equally important in this scenario, i.e., the weight factor of each indicator is 0.2, then the intrusion detection data security score is:

$$0.2 \times 95.06\% + 0.2 \times 99.92\% + 0.2 \times 99.95\% + 0.2 \times 80.65\% + 0.2 \times 82.35\% = 91.586\%.$$

Method 2: The AHP can combine the qualitative information such as the decision maker's empirical judgement with the quantitative information such as the data, so as to make the decision-making results more scientific and reasonable. Next, the AHP will be used to calculate the weight of each evaluation index, so as to evaluate the security of intrusion detection data in a more reasonable and comprehensive way.

First, the subjective degree of scoring based on the importance of each security indicator is used to obtain a judgement matrix C:

$$C = \begin{bmatrix} 1, & 0.2, & 0.143, & 3, & 6 \\ 5, & 1, & 0.33, & 7, & 9 \\ 7, & 3, & 1, & 6, & 9 \\ 0.33, & 0.143, & 0.167, & 1, & 2 \\ 0.167, & 0.11, & 0.11, & 0.5, & 1 \end{bmatrix}$$

After solving, the maximum eigenvalue is 5.29 and the number of non-zero eigenvalues is 5, which also means that the order of the matrix is 5. By querying Table 5, when the matrix order is 5, the value of RI is 1.12. Thus, consistency checking can be done next:

$$CR = CI/RI = \left(\frac{5.29 - 5}{5 - 1} \right) / 1.12 = 0.06 < 0.1,$$

where the judgement matrix C satisfies the consistency requirement. The maximum vectors corresponding to the maximum eigenvalue are [0.2131, 0.4208, 0.8757, 0.888, 0.0519]. Furthermore, after normalizing the eigenvector of the largest eigenvalue, the weight of each indicator is [0.087, 0.172, 0.358, 0.363, 0.02]. Hence, the security of the intrusion detection dataset is:

$$\begin{aligned}
\Omega_D &= W_{D1}S_{DC} + W_{D2}S_{DA} + W_{D3}S_{DCO} + W_{D4}S_{DD} + W_{D5}S_{DB} \\
&= 0.087*95.06\% + 0.172*99.92\% + 0.358*99.95\% + 0.363*80.65\% + 0.02*82.35\% \\
&= 92.17\%
\end{aligned}$$

The security score of the data is 92.17%, using the AHP method for evaluation. Compared with the mean evaluation method, although the final evaluation results are not significantly different, the AHP method can more accurately identify core indicators such as accuracy, consistency and diversity, by assigning them higher weights, which makes the final evaluation results more in line with the actual data security situation. Moreover, by focusing on the core indicators, it can provide clearer guidance on data security improvement for researchers.

A.2 Model security evaluation for intrusion detection

In the field of intrusion detection, deep learning leverages its ability for layer-wise nonlinear transformations to automatically mine complex patterns and deep-seated features within intrusion detection data, thereby enhancing the accuracy of anomaly detection. Given that intrusion detection data in real-world network environments is often accompanied by noise and incompleteness, the stacked denoising autoencoder (SDA) demonstrates certain robustness against data noise. This use case will train an intrusion detection model based on the SDA algorithm. However, during SDA training, if the model learns excessive details and noise from the training data – particularly when training data volume is limited – it may lead to performance degradation in detecting new intrusion samples, compromising the reliability of its output. Therefore, the security of the model requires evaluation.

A.2.1 Determination of evaluation indicators of model security for model security

Models with high accuracy can effectively identify various types of intrusion behaviours, reducing resource waste caused by false positives and potential security risks arising from false negatives. Therefore, model accuracy serves as a key indicator for measuring its fundamental usability. Given that intrusion detection operates in complex, volatile and uncertain network environments where attack methods rapidly evolve and continuously change, robustness evaluation ensures the model maintains stable performance when confronted with disturbances and evolving attack techniques. Credibility evaluation focuses on determining the trustworthiness of the model's output, enabling security personnel to make more precise judgements regarding the authenticity of detection alerts. Interpretability evaluation helps analyse how the model makes decisions based on input data, providing a solid foundation for security personnel to formulate subsequent protective measures. Thus, to comprehensively evaluate model security and ensure its effectiveness in dynamic intrusion detection scenarios, we conduct an in-depth evaluation across four dimensions: accuracy, robustness, credibility and interpretability.

A.2.2 The security indicators evaluation of model security for model security

The evaluation results of security indicators for the intrusion detection model are summarized in Table A.3. A more detailed explanation of each evaluation indicator will be provided subsequently.

Table A.3 – The evaluation results of security indicators for the intrusion detection model

Indicator	Accuracy	Robustness	Credibility	Interpretability
Score	82.48%	80.89%	91.15%	50%

Accuracy: In intrusion detection, data often exhibits severe class imbalance, with normal traffic data being abundant while intrusion behaviour data is relatively scarce. In such cases, the accuracy may appear high due to the correct classification of numerous normal samples, which may mask the model's true capability to detect intrusion samples. In practice, for intrusion detection, the

consequences of false negatives are typically more severe than those of false positives. This is because false negatives imply that intrusion behaviours go undetected, potentially leading to system breaches, data leaks and other critical security incidents. The core objective of the recall rate is to minimize undetected intrusion samples. A higher recall rate indicates fewer missed intrusion samples, reflecting the model's stronger capability to identify security threats and better aligning with practical security requirements. Therefore, the recall rate is adopted to evaluate the effectiveness of the intrusion detection model. After validation, the model's recall rates for the eight categories (normal, DoS, DDoS, WEb attack, infiltration, malware attack, brute force attack and PortScan) are 95.47%, 91.28%, 81.7%, 75.16%, 57.21%, 80.44%, 85.09%, and 93.49%, respectively. Consequently, the model's accuracy is calculated as follows:

$$(95.47 + 91.28 + 81.7 + 75.16 + 57.21 + 80.44 + 85.09 + 93.49) / 8 * 100\% = 82.48\%.$$

Robustness: Validation set: Add 20% white noise randomly. The adversarial classification accuracy (SM_ACA) is adopted to verify the model's correct classification rate for noisy samples, thereby evaluating the model's robustness. The total number of samples in the test set is 966 229, of which 193 246 are noisy samples. After testing, the number of noise samples correctly classified by the model is 156 329. Thus, the robustness score of the model is calculated as:

$$156\ 329 / 193\ 246 * 100\% = 80.89\%.$$

Credibility: The model's average output confidence scores for the eight categories (normal, DoS, DDoS, web attack, infiltration, malware attack, brute force attack and PortScan) are 0.86, 0.84, 0.76, 0.65, 0.54, 0.70, 0.74 and 0.80, respectively. The recall rates are 95.47%, 91.28%, 81.7%, 75.16%, 57.21%, 80.44%, 85.09% and 93.49%, respectively. The absolute differences between these values are 9.47%, 7.28%, 5.7%, 10.16%, 3.21%, 10.44%, 11.09% and 13.49%, respectively. The model's credibility score is:

$$\frac{1}{8} (1 - 9.47\%) + (1 - 7.28\%) + (1 - 5.7\%) + (1 - 10.16\%) + (1 - 3.21\%) + (1 - 10.44\%) + (1 - 11.09\%) + (1 - 13.49\%) * 100\% = 91.15\%$$

Interpretability: The denoising autoencoder consists of an encoder and a decoder. The decoder reconstructs low-dimensional features into the original input data. By analysing the reconstruction effectiveness of the decoder – i.e., comparing the differences between the original data and the reconstructed data – the features learned by the model and its ability to reconstruct input data can be understood. This further explains how the model compresses and represents data while preserving essential information. Consequently, the interpretability capability of the SDA scores 50 points.

A.2.3 Comprehensive evaluation for model security

Based on the evaluation scores of the above four indicators, the comprehensive evaluation of model security is conducted as follows.

Evaluation method 1: If all security evaluation indicators are equally important (i.e., each indicator's weight coefficient is 0.25), the security score of the intrusion detection model is calculated as: $0.25 * 82.48\% + 0.25 * 80.89\% + 0.25 * 91.15\% + 0.25 * 50\% = 76.13\%$.

Evaluation method 2: The analytic hierarchy process (AHP) method is adopted for model security evaluation. The judgement matrix is structured as follows:

$$C = \begin{bmatrix} 1, & 1, & 1, & 7 \\ 1, & 1, & 0.33, & 6 \\ 1, & 3, & 1, & 8 \\ 0.143, & 0.167, & 0.125, & 1 \end{bmatrix}$$

After solving, the maximum eigenvalue is 4.12 and the number of non-zero eigenvalues is 4. By referencing the table, when the matrix order is 4, the value of RI is 0.9. The consistency check result is calculated as follows:

$$CR=CI/RI=(\frac{4.12-4}{4-1})/0.9=0.04 < 0.1,$$

The judgement matrix C satisfies the consistency requirement. Furthermore, the eigenvector corresponding to the maximum eigenvalue is $[0.53, 0.39, 0.74, 0.08]$. After normalization, the weights for each indicator are $[0.32, 0.23, 0.44, 0.01]$. Therefore, the security score of the intrusion detection model is calculated as:

$$\begin{aligned}\Omega_D &= 0.32*82.48\% + 0.23 * 80.89\% + 0.44*91.15\% + 0.01 * 50\% \\ &= 85.6\%\end{aligned}$$

The mean method typically involves simple averaging calculations across various indicators, assuming equal importance for all indicators. However, this approach often fails to align with real-world complexities in many evaluation scenarios. In the field of cybersecurity, particularly within the critical context of intrusion detection, models must demonstrate exceptional robustness to ensure highly stable and reliable outputs, enabling them to calmly handle diverse unknown scenarios. Simultaneously, models require strong self-awareness capability to maintain a clear and explicit understanding of their own capability boundaries. Compared to the mean method, the analytic hierarchy process (AHP) offers distinct advantages: it flexibly prioritizes important indicators based on specific scenario requirements and thoroughly analyses the significance and interrelationships among factors, thereby better fulfilling practical application evaluation requirements.

A.3 Environment security evaluation for intrusion detection

The trained SDA-based intrusion detection system (IDS) is deployed behind the firewall for application testing, with key components including GPU RTX 3090, Linux (Ubuntu) operating system, Anaconda programming environment, Python 3.12 programming language, TensorFlow deep learning framework and data processing/analysis libraries such as NumPy (numerical computing), Pandas (data processing), Matplotlib and Seaborn (data visualization).

A.3.1 Determination of evaluation indicators of environment security for environment security

The intrusion detection system (IDS) exhibits high dependency on software and hardware environments. Evaluating its vulnerability enables proactive vulnerability remediation, thereby enhancing system resilience. Compromised environmental completeness may introduce biases in data collection and analysis, leading to omissions of intrusion behaviours. The network environment is complex, the attack intensity is high and the scale is large. Evaluating the robustness can test the system's ability to resist and recover from stress, discover the weak links and enhance the protection effect.

A.3.2 The security indicators evaluation of environment security for environment security

Vulnerability: According to the evaluation items of environment vulnerability, based on this actual deployment of IDS, they are evaluated separately, and the evaluation results are summarized in Table A.4. Therefore, after summation, the vulnerability score is 95%.

Table A.4 – The environmental vulnerability evaluation scores

Category	Evaluation item	Description	Score (%)	State
Hardware	Debugging interface security	Strict debugging interface security mechanisms have been established and implemented.	5	☑
	Chip security	Strict chip security mechanisms have been established and implemented.	5	☑
	Trusted computing	Trusted computing has been implemented.	3	☒
Software	Vulnerability patching	Ensure that there are no high-risk vulnerabilities related to the software published two months ago.	8	☑
		Software exception monitoring has been implemented.	3	☑
Operating system	Vulnerability patching	An automatic and mandatory upgrade mechanism has been established to eliminate security vulnerabilities.	8	☑
		Ensure that there are no high-risk vulnerabilities related to the system published two months ago.	8	☑
		Effective system exception monitoring has been implemented to promptly identify issues.	8	☑
	Permission restriction	The principle of least privilege is adhered to. Users only have the minimum permissions required to perform their tasks and all unauthorized permissions are prohibited.	3	☑
	Boot security	System secure boot authentication and validation strategies have been set.	3	☑
		System secure boot authentication and validation strategies are effective.	3	☑
	Update security	The system supports both manual and automatic updates.	5	☑
		Before upgrading, version checking, MD5 checking and signature checking of the image are performed.	5	☑
		When system updates are turned off or fail, it can roll back to the pre-update version.	3	☑
	Configuration security	System partition configuration security measures have been implemented.	3	☑
		Default closing of debugging ports, default closing of remote login ports and default closing of debugging log information services.	4	☑
		Reasonable security attributes have been established and configured for input and output ports.	3	☑
		Access control mechanisms have been implemented to prevent unauthorized calls to system interfaces.	3	☑
		Login passwords meet complexity requirements.	5	☑
Code	Code audit	Key codes have passed security audits.	5	☑

Completeness: According to the evaluation items of environment completeness, the evaluation results are summarized in Table A.5. Therefore, after summation, the vulnerability score is 90%.

Table A.5 – The environmental completeness evaluation scores

Category	Evaluation item	Description	Score (%)	State
Application data	Secure encryption	Important data in logs is encrypted.	5	☑
		Sensitive data such as database connection passwords, FTP service passwords, login passwords and external system interface authentication passwords stored in context files (including configuration files, databases, cookies, etc.) during the application process are encrypted.	5	☑
	Secure authentication	Resistant to brute – force attacks.	5	☑
		Security checks and risk evaluations have been implemented.	8	☑
	Access check	Strict access-role checking mechanisms have been implemented.	3	☑
	Security protection	It has the ability to resist malicious tampering and malicious analysis.	8	☑
	Transmission security	Data transmission security measures have been implemented to ensure the confidentiality, integrity and validity of data.	8	☑
	Storage security	Important data is stored encrypted.	8	☑
		Important data is backed up, and the backup mechanism meets security requirements.	3	☑
		Important data is stored in dedicated data hosts or storage media.	3	☑
		Between device identification information, device configuration data, device parameter information, operating system data, and application software data, at least logical isolation from other categories of data has been implemented.	3	☑
	Access control security	Unauthorized entity data access can be terminated and perceptible alarm functions are provided.	5	☑
		Access to or attempts to modify data are knowable and controllable.	5	☑
Communication Process	HTTPS filtering security	Blacklist filtering and interception of HTTPS have been implemented.	3	☑
		Strategy cache management has been implemented.	3	☑
	IP filtering security	IP blacklists and whitelists have been set. IP source or destination addresses in IP data packets forwarded by routers are filtered.	4	☑
		Support for filtering of TCP, UDP and ICMP protocols.	3	☑

Table A.5 – The environmental completeness evaluation scores

Category	Evaluation item	Description	Score (%)	State
	Blacklist and whitelist management security	Cloud-based blacklist and whitelist management has been implemented.	3	☑
	Cloud database malicious interception	Cloud-based HTTPS query and interception of malicious databases have been implemented.	5	☑
	Trusted boot and static measurement	Trusted boot functions have been implemented.	5	☒
		File static measurement has been implemented, including elf files, so files, kernel module ko files, sh files and config files.	5	☒

Robustness: The robustness of the IDS system is evaluated by limiting network bandwidth and simulating network interruption situations. After simulating 1000 times, the model was able to quickly adjust and complete the detection task 618 times. Therefore, the robustness score is $618/1000 \times 100\% = 61.8$.

A.3.3 Comprehensive evaluation for environment security

The overall evaluation of environment security is conducted based on the vulnerability, completeness and robustness. In practice, the accuracy of the final evaluation results is likely to be affected by personal bias or misunderstanding during the evaluation process. Therefore, we use FAHP to measure the environmental security more realistically through the input of 10 experts in this case. The results of the expert consultation are summarized in Table A.6.

Table A.6 – The results of the expert consultation

Comparison scale	1	2	3	4	5	6	7	8	9
Vulnerability and completeness	4	2	4	–	–	–	–	–	–
Vulnerability and robustness	–	–	2	4	4	–	–	–	–
Completeness and robustness	–	–	–	2	3	5	–	–	–

According to Table A.6, it can be seen that the comparison between vulnerability and completeness, four experts considered both equally important, four considered vulnerability to be slightly more important than completeness, and the rest of the experts considered it to be somewhere in between. Thus, the quantized value of vulnerability and completeness. Therefore, the comparison scale of vulnerability and completeness is calculated as follows:

$$\Omega_D = 1 * \frac{4}{10} + 2 * \frac{2}{10} + 3 * \frac{4}{10} = 2$$

In this manner, the remaining comparison scales are computed sequentially, and the fuzzy judgement matrix is:

$$C_1 = \begin{bmatrix} 1, & 2, & 4.2 \\ 0.5, & 1, & 5.3 \\ 0.24, & 0.19, & 1, \end{bmatrix}$$

The maximum eigenvalue is 3.08, the consistency check result is:

$$CR=CI/RI=(\frac{3.08-3}{3-1})/0.58=0.07 < 0.1,$$

The judgement matrix C_1 satisfies the consistency requirement. After normalization, the weights for each indicator are [0.54, 0.37, 0.09]. Therefore, the security score of the intrusion detection model is calculated as:

$$\Omega_D = 0.54*95\% + 0.37*90\%+0.09*61.8\% = 90.16\%$$

A.4 Comprehensive evaluation for intrusion detection system

Based on the above evaluation, the weights assigned to data, model and environment are equivalent. Consequently, the final security score of IDS is calculated as follows:

$$\begin{aligned}\Theta_{sys} &= 0.33*92.17\% + 0.33 * 85.6\%+0.33*90.16\% \\ &= 88.41\%\end{aligned}$$

The security of this IDS is 88.41%. Quantitative security evaluation of the IDS objectively and effectively proves the reliability of the system, thus exceeding the subjective "feeling of security" and providing a quantitative view of the real security risks, which strengthens the user's confidence in the system and improves the willingness to deploy the system to protect the critical information infrastructure (CII). In addition, the quantitative assessment provides a reliable validation basis and optimization direction for model updating and iteration (retraining, parameter tuning, architectural optimization), reduces the cost of trial and error and ensures that the system is able to robustly adapt to the complex and ever-changing cyber threat environment.

Bibliography

- [[b-ITU-T Y.3090](#)] Recommendation ITU-T Y.3090 (2022), *Digital twin network – Requirements and architecture*.
- [b-ISO 9000] ISO 9000:2015, *Quality management systems – Fundamentals and vocabulary*.
- [b-ISO/IEC 42001] ISO/IEC 42001:2023, *Information technology – Artificial intelligence – Management system*.
- [b-ISO/IEC TS 5723] ISO/IEC TS 5723:2022, *Trustworthiness – Vocabulary*.
- [b-NIST AI RMF 1.0] Tabassi, E. (2023), *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*, NIST AI 100-1, Gaithersburg, MD, National Institute of Standards and Technology, doi: 10.6028/NIST.AI.100-1.
- [b-Elouataoui] Elouataoui, W., Elmendili, S., and Gahi, Y. (2023), *Quality Anomaly Detection Using Predictive Techniques: An Extensive Big Data Quality Framework for Reliable Data Analysis*, IEEE Access, Vol. 11, pp. 103306–103318, doi: 10.1109/ACCESS.2023.3317354.
- [b-GB/T 43782-2024] State Administration for Market Regulation and Standardization Administration of China (2024), GB/T 43782-2024, *Artificial intelligence – Technical requirements for machine learning system*.
- [b-Guo] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K.Q. (2017), *On Calibration of Modern Neural Networks*, Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research, Vol. 70, PMLR, pp. 1321–1330.
- [b-Han] Han, C., Sun, J., Bian, Y., Que, W., and Shi, L. (2023), *Automated Detection and Localization of Myocardial Infarction With Interpretability Analysis Based on Deep Learning*, IEEE Transactions on Instrumentation and Measurement, Vol. 72, pp. 1–12, doi: 10.1109/TIM.2023.3258521.
- [b-Li] Li, A., Zhang, L., Tan, J., Qin, Y., Wang, J., and Li, X.-Y. (2021), *Sample-level Data Selection for Federated Learning*, IEEE INFOCOM 2021 – IEEE Conference on Computer Communications, Vancouver, BC, Canada, IEEE, pp. 1–10, doi: 10.1109/INFOCOM42981.2021.9488723.
- [b-Paul] Paul, S., Vijayshankar, S., and Macwan, R. (2024), *Demystifying Cyberattacks: Potential for Securing Energy Systems With Explainable AI*, 2024 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, IEEE, pp. 430–434, doi: 10.1109/ICNC59896.2024.10556212.
- [b-Reif] Reif, E., Kahng, M., and Petridis, S. (2023), *Visualizing Linguistic Diversity of Text Datasets Synthesized by Large Language Models*, 2023 IEEE Visualization and Visual Analytics (VIS), Melbourne, Australia, IEEE, pp. 236–240, doi: 10.1109/VIS54172.2023.00056.
- [b-Saaty] Saaty, T.L. (1980), *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, New York, McGraw-Hill International Book Company.

- [b-Shannon] Shannon, C.E. (1948), *A Mathematical Theory of Communication*, The Bell System Technical Journal, Vol. 27, No. 3, July, pp. 379–423, doi: 10.1002/j.1538-7305.1948.tb01338.x.
- [b-Sun] Li, Z., Sun, J., Yang, K., and Xiong, D. (2022), *A Review of Adversarial Robustness Evaluation for Image Classification*, *Journal of Computer Research and Development*, Vol. 59, No. 10, pp. 2164–2189, doi: 10.7544/issn1000-1239.20220507.
- [b-Tian] Tian, H.X., Wang, H.G., and Zhou, K. (2018), *Data Quality Assessment for On-line Monitoring and Measuring System of Power Quality Based on Big Data and Data Provenance Theory*, 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), IEEE, pp. 248–252, doi: 10.1109/ICCCBDA.2018.8386521.
- [b-Vidakovic] Vidakovic, B. (2011), *Statistics for Bioengineering Sciences: With MATLAB and WinBUGS Support*, Springer Texts in Statistics, New York, Springer, ISBN: 978-1-4614-0393-7.
- [b-Zituo] Li Zituo, Sun Jianbin, Yang Kewei, Xiong Dehui (2022), *A Review of Adversarial Robustness Evaluation for Image Classification*, *Journal of Computer Research and Development*, vol. 59, no. 10, pp. 2164-2189, doi: 10.7544/issn1000-1239.20220507.
- [b-Wu] Wu, T., Chen, L., Hui, P., Zhang, C.J., and Li, W. (2015), *Hear the Whole Story: Towards the Diversity of Opinion in Crowdsourcing Markets*, *Proceedings of the VLDB Endowment*, Vol. 8, No. 5, pp. 485–496, doi: 10.14778/2735479.2735482.
- [b-XSTR-SEC-AI] ITU-T Technical Report XSTR-SEC-AI (2022), *Guidelines for security management of using artificial intelligence technology*, International Telecommunication Union.
-