Translation

The following document is a draft Chinese national standard for the security of machine learning (ML) algorithms. Drafted by Chinese government entities and tech companies, the standard provides security guidelines for developers of ML models and procedures for assessing ML algorithms' susceptibility to cyberattacks.

Title
Information security technology-Security specification and assessment methods for machine learning algorithms
信息安全技术 机器学习算法安全评估规范

Source
National Information Security Standardization Technical Committee (全国信息安全标准化技术委员会) website. This draft standard is dated July 27, 2021 and was uploaded to the website on August 4, 2021.

The Chinese source text is available online at:
https://www.tc260.org.cn/file/2021-08-04/6b530404-858b-4c9d-8d89-a83239ec5712.pdf
An archived version of the Chinese source text is available online at: https://perma.cc/Y5WL-56NJ

Translation Date	**Translator**	**Editor**
February 28, 2023	Etcetera Language Group, Inc.	Ben Murphy, CSET Translation Manager

National Standard of the People's Republic of China

Information security technology-Security specification and assessment methods for machine learning algorithms

(Draft for Comments)

(Draft Completed on: July 27, 2021)

When submitting feedback, please include relevant patents that you are aware of along with supporting documentation.

State Administration for Market Regulation
Standardization Administration of China

Issuers

Contents

Preface

This document is drafted in accordance with the provisions of GB/T 1.1-2020 *Directives for standardization -- Part 1: Rules for the structure and drafting of standardizing documents*.

This document is proposed and administered by the National Information Security Standardization Technical Committee (SAC/TC 260).

Drafting organizations of this document: Beijing CESI Technology Co., Ltd., Tsinghua University, Beijing RealAI Intelligent Technology Co., Ltd. (北京瑞莱智慧科技有限公司), the National Research Center for Information Technology Security (国家信息技术安全研究中心), Guangzhou University, the National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC; 国家计算机网络应急技术处理协调中心), Huawei Technologies Co., Ltd., Beijing Megvii Technology Co., Ltd., China Academy of Information and Communications Technology (CAICT), Beijing Baidu Netcom Science and Technology Co., Ltd., Institute of Information Engineering – Chinese Academy of Sciences (CAS), Alibaba (Beijing) Software Services Co., Ltd., Shenzhen Tencent Computer Systems Co., Ltd., Beijing Qihoo Technology Co., Ltd., Chongqing University of Posts and Telecommunications, Shenzhen Research Institute of Big Data, Beijing Research Institute for Computer Technology and Applications (北京计算机技术及应用研究所), and China Electronics Standardization Institute (CESI).

Drafted by: Shangguan Xiaoli, Hu Ying, Hao Chunliang, Zhang Yuguang, Su Hang, Hu Songzhi, Yang Tao, Jing Huiyun, Zhang Xudong, Xu Xiaogeng, Gu Zhaoquan, Wu Yuesheng, Meng Guozhu, Li Shi, Fu Yingbo, Mei Jingqing, Wang Le, Dong Yinpeng, Liu Xize, Wang Zhelin, Zhao Yunwei, Han Han, Zhang Xia, Peng Juntao, Xu Yongtai, Zhang Yi, Xu Yuqing, Wu Baoyuan, Han Lei, and Wang Bingzheng.

Information security technology - Security specification and assessment methods for machine learning algorithms

1 Scope

This document specifies the security requirements and confirmation (证实) methods for machine learning algorithms used in design and development, verification and testing, deployment and operations, maintenance and upgrading, decommissioning and removal, and other stages as well as the implementation of security assessments for machine learning algorithms.

This document is applicable to the security assessment of algorithms in machine learning systems. It also applies to machine learning system developers and operators conducting security measure self-assessment and enhancement in the process of algorithm development and operation.

2 Normative References

The contents of the following documents, through normative references in this text, constitute indispensable provisions of this document. Among them, for dated references, only the edition corresponding to that date applies to this document. For undated references, the latest edition (including all amendments) applies to this document.

GB/T 25069—2010	Information security technology	Glossary
GB/T 35273—2020	Information security technology	Personal information security specification
GB/T 37988—2019	Information security technology	Data security capability maturity model

3 Terms and Definitions

The terms and definitions defined in GB/T 25069-2010 and the following ones apply to this document.

3.1 Adversarial examples (对抗样本)

Input examples formed by deliberately adding subtle interference to the dataset. Such examples may cause the model to give false outputs with high confidence.

3.2 Personal information

Any information recorded using electronic or other means that can be used alone or in combination with other information to identify a natural person or reflect the special characteristics of the activities of a natural person.

[Source: GB/T 35273-2020, Terms and Definitions 3.1]

Note: See GB/T 35273-2020 for the scope and types of personal information.

3.3 Machine learning algorithm

These algorithms use machine learning technologies and theories to solve problems. They clearly define a limited and ordered set of rules and generate classifications, reasoning, predictions, and other outputs based on input data.

3.4 Machine learning algorithm life cycle

The entire evolutionary process of a machine learning system algorithm from inception to decommissioning, including design and development, verification and testing, deployment and operation, maintenance and upgrading, and decommissioning and removal.

Note: During the life cycle of a machine learning algorithm, some activities can occur in different processes, and individual processes can be repeated. For example, bug fixes and updates require performing the development and deployment processes repeatedly.

3.5 Algorithm failure

An event in which an algorithm fails to complete its prescribed function.

3.6 Outliers

Data in a sampled dataset that significantly deviates from the trend presented by the majority of the data.

3.7 Accuracy

For a given dataset, the ratio of the number of correctly classified examples to the total number of examples.

4 Acronyms

The following abbreviations apply to this document.
SDK: Software Development Kit

5 Overview

Machine learning algorithms have the characteristics of needing to learn from data or experience, uncertain output results, and unexplainable (不可解释) decision-making processes. Therefore, the security of machine learning algorithms involves not only the security of the algorithm itself, but also the security of algorithm applications. All stages in the machine learning algorithm life cycle are faced with security risks at the algorithm, data, and environmental levels. In all scenarios, machine learning algorithm security

needs to satisfy basic security attributes such as confidentiality (保密性), integrity, availability, controllability, robustness, and privacy. In important fields such as public services, transportation and driving, financial services, health and hygiene, and welfare and education (福利教育), the security requirements for machine learning algorithms deployed for decision-making related to key matters such as life and property safety and protection of individual rights (个人权利) are stricter.

In this document, the security requirements for machine learning algorithms are divided into basic level and enhanced level:

— Basic level: The text used for security requirements clauses is not bold, indicating the machine learning algorithm security requirements applicable to all scenarios.

— Enhanced level: The text used for security requirements clauses is bold, indicating additional security requirements for machine learning algorithms used for decision-making related to key matters such as life and property safety and protection of individual rights in important fields such as public services, transportation and driving, financial services, health and hygiene, and welfare and education.

The confirmation methods put forward in Chapter 7 of this document have a one-to-one correspondence with the security requirements put forward in Chapter 6. In Chapter 7, the confirmation methods that correspond to enhanced security requirements in Chapter 6 are indicated by bold font.

Note 1: See Appendix A for the security attributes and security index system for machine learning algorithms.

Note 2: See Appendix B for the security risks of machine learning algorithms.

Note 3: As a characteristic of machine learning algorithms is their need to learn from data or experience, algorithm security must include the security of training data and test data. In particular, machine learning algorithms that involve personal information also need to meet privacy requirements.

6 Security Requirements

6.1 General

General security requirements for machine learning algorithms include:

a) When organizations and individuals carry out data processing activities related to machine learning algorithms, they shall possess basic data security capabilities, which shall reach at least level 2, and preferably level 3, as specified in GB/T 37988-2019.

b) When developing or operating machine learning algorithms, relevant

organizations and individuals shall ensure the confidentiality of machine learning algorithms developed or operated and ensure that the machine learning algorithm models, data, and dependency information (依赖信息) are not leaked to unauthorized individuals, entities, or processes. Relevant measures include but are not limited to:

1）In the verification and testing stage, the algorithm shall pass tests on the prevention of membership inference attacks and [model] inversion attacks.
2）Data and model destruction shall ensure that deleted data and models cannot be recovered.
3）Set access permissions to deny access to unauthorized entities.

Note: See Appendix A.1 for confidentiality indicators.

c) When developing or operating machine learning algorithms, relevant organizations and individuals shall ensure the integrity of machine learning algorithms developed or operated and ensure that the machine learning algorithm models, data, and software and hardware dependency information are not replaced or destroyed by unauthorized means. Relevant measures include but are not limited to:

1） In the algorithm design and development and application deployment stages, perform regular verification of data, model, and runtime environment to prevent the data, model, and runtime environment from being tampered with.
2）Conduct tests on indicators such as dataset size, balance, accuracy, and relevance to algorithm tasks to ensure that they meet algorithm requirements.

Note: See Appendix A.2 for integrity indicators.

d) When developing or operating machine learning algorithms, relevant organizations and individuals shall ensure the availability of machine learning algorithms developed or operated and ensure that when authorized users need to do so, they can access and use machine learning algorithm models, data, and dependency information. Relevant measures include but are not limited to:

1） Fix publicly disclosed (公开) vulnerabilities and prevent attacks such as penetration attacks (渗透攻击) from interfering with the normal operation of the software and hardware upon which algorithm operation depends.
2）In the design documentation, add restrictions for attributes such as data formats and size.

3）Test the ability to quickly return to the operating state after an algorithm failure event occurs, including the recovery time and degree of recovery.
4）Use a method combining sampling tests and full tests to check whether the data can accurately represent the actual objects it describes.

Note: See Appendix A.3 for availability indicators.

e) When developing or operating machine learning algorithms, relevant organizations and individuals shall ensure the controllability of the developed or operated machine learning algorithms, ensure that their specified functions are completed under specified conditions and within the specified time, and that they provide a mechanism that allows algorithm operation behaviors to be taken over many an operator in the case of improper machine learning algorithm behavior. Relevant measures include but are not limited to:
   1）Under the specified time and conditions, test the number of algorithm failures and their severity and frequency, calculate the proportion of failures resolved, and ensure that the continuous operating time of the algorithm meets the set threshold.
   2）During the deployment and operation stage, record the normal service time and cumulative effective service time of the algorithm and calculate the normal service time proportion.
   3）During the algorithm life cycle, according to the requirements of each stage of project management, compile documentation materials and log records of the algorithm's key decision-making points to provide auditability and traceability.

Note: See Appendix A.4 for controllability indicators.

f) When developing or operating machine learning algorithms, relevant organizations and individuals shall ensure the robustness of machine learning algorithms developed or operated and ensure that the machine learning algorithm system can maintain its performance level under any circumstances. Relevant measures include but are not limited to:
   1）Use lossy data such as compressed and corrupt data and interference data such as data with added noise or that has been transformed to test the correctness (正确性) of algorithm functions.
   2）In white-box and black-box adversarial scenarios, test the algorithm accuracy and whether it meets the set threshold.
   3）Carry out adversarial example attack, physical adversarial attack, and backdoor attack tests on the algorithm to ensure that the attack success rate is lower than the set threshold.

4）Set thresholds for model feedback output, query counts, and query frequency in the design documentation to avoid access control attacks.

Note 1: See Appendix A.5 for robustness indicators.

Note 2: See Appendix C for adversarial example attacks.

g) When developing or operating machine learning algorithms, relevant organizations and individuals shall ensure the privacy of machine learning algorithms developed or operated, ensure that data processing complies with legal and regulatory requirements, protect personal information and privacy, and avoid storing and leaking sensitive data. Relevant measures include but are not limited to:

1）Personal information shall not be used for machine learning algorithm-related activities without the consent of the individual, except in cases where consent is not required by laws and regulations.

2）Adopt necessary data masking (数据脱敏) measures for personal information.

Note 1: Personal information includes faces, voiceprints, and ID numbers. See GB/T 35273-2020.

Note 2: See Appendix A.6 for privacy indicators.

6.2 Design and Development Stage

When carrying out design and development, relevant organizations or individuals shall:

a) Test the security of the data used, repair or filter out data detected to be contaminated, and keep detection and disposal records.

Note: Detection includes data morphing (数据变形) detection, forgery detection, and backdoor detection.

b) According to use case security requirements, analyze and determine the algorithm availability indicators, save analysis records, and ensure that algorithm development conforms to the indicator settings.

c) According to use case security requirements, analyze and determine the following training data indicators, save analysis records, and ensure that the training data meets the indicator requirements:

1） Training data scale threshold;

2） Training data balance indicator;

3） Training data labeling accuracy threshold.

Note: The accuracy can be measured by multi-source label cross-validation and other methods.

d) Design an emergency response mechanism to ensure that algorithm operation can be interrupted when necessary.

e) Carry out activities related to improving algorithm robustness, including but not limited to:

1） Use methods such as adversarial training to improve robustness.

2） Record the robustness improvement process, including information such as the time required and important operations,

3） Evaluate the effect in improving algorithm robustness and draft an assessment report.

6.3 Verification and Testing Stage

When carrying out verification and testing, relevant organizations or individuals shall:

a) Adopt a method combining dynamic testing and static testing to detect and locate algorithm defects, backdoors, and potential risks.

b) If a third party is entrusted to carry out testing, they must possess security mechanisms for algorithm and data confidentiality. The methods that can be adopted include but are not limited to:

1） Carry out testing using the environment and equipment provided by the algorithm owner.

2） Use two or more third parties to separately test the same algorithm on different data types.

c) Based on test tasks, analyze and set the repeatability index (重复度指标) between test data and training data, save the analysis process, and ensure that the test data conforms to the index setting.

d) Include reproducibility testing:

1） They shall pre-analyze and set the reproducibility threshold according to the security requirements of the use case and the test situation and retain the analysis process.

2） They shall carry out verification tests on algorithm reproducibility and record the test results.

Note: Reproducibility refers to the ability of an algorithm to produce the same or highly similar results when processing the same data.

e) According to test tasks, analyze and determine the following indicators,

save analysis records, and ensure that the test data meets the indicator requirements:

1）Test data scale threshold;

2）Test data balance indicator;

3）Test data labeling accuracy threshold;

4）Test data and test task correlation threshold.

f) According to test tasks, fully verify the robustness of the test algorithm, including the use of data that contains natural noise and data that is fabricated, counterfeit, random, meaningless, or irrelevant to the application scenario of the algorithm.

g) Properly simulate improper algorithm behavior scenarios to check whether the algorithm provides methods or measures for manual takeover and termination.

6.4 Deployment and Operation Stage

When carrying out deployment and operation, relevant organizations or individuals shall:

a) Add restrictions on attributes such as input data format and size to prevent unusual input data from causing model errors. In scenarios where there are many interfering inputs, mechanisms such as input screening and filtering shall be added to ensure the stable operation of the algorithm.

Note: Interfering inputs include outlier data.

b) Protect private data such as internal model parameters by limiting the model feedback output, limiting the number of model queries, and limiting the usage frequency of accounts and IPs so as to prevent attackers from backward chaining (逆向推测) and restoring model parameters and training data.

c) Take measures to reduce the risk of reverse engineering of algorithm parameter files and code files, including encrypted storage algorithms and algorithm code obfuscation.

d) Describe the functions, limitations, security risks, and possible effects of machine learning algorithms to users in a timely, accurate, complete, clear, and unambiguous manner, and explain the relevant application process and application results.

e) Possess a data security protection mechanism to ensure the confidentiality, integrity, and availability of data. The methods that can be adopted include but are not limited to:

1） Encryption algorithms;

2）Integrity verification.

f) Set up emergency response mechanisms, including manual emergency intervention and termination when algorithms cause security problems.

g) If an unexplainable algorithm is deployed, it shall only be used as an auxiliary decision-making method, not as a direct basis for decision-making.

6.5 Maintenance and Upgrading Stage

When carrying out maintenance and upgrades, relevant organizations or individuals shall:

a) If the algorithm is significantly adjusted, modified, or upgraded, promptly update the model parameters and configuration and delete irrelevant parameters and data to ensure that the change process meets the security requirements of the design and development, verification and testing, and deployment and operation stages. In addition, they shall keep records on algorithm changes, including at least the time and description of the algorithm change, as well as operation records on the corresponding updating and deletion of model parameters and configurations.

b) Set up a security verification mechanism to carry out security verification on the model upgrade package file and ensure the security of the upgrade package. In addition, they shall keep records on model upgrade verification, including, at a minimum, the model upgrade verification time, version, and key verification information operation records.

6.6 Decommissioning and Removal Stage

When carrying out decommissioning and removal, relevant organizations or individuals shall:

a) Set the conditions to be met for decommissioning and removal, and set a reasonable time period for the destruction of data and models.

b) Destroy the data in storage media and relevant documentation materials, including training data, test data, instance data, feature data, parameters, and algorithm outputs. Feature data, parameters, and algorithm outputs do not have to be destroyed if the following conditions are simultaneously met:

1）It is necessary for the further realization of actual business functions.

2）Explicit authorization and consent are obtained from the data owner.

3）No personal information or important data as identified by relevant departments are involved.

4）After obfuscation or encryption, it is not possible to directly restore the original data.

c) Ensure that destroyed models can no longer be accessed.

d) If personal information is de-identified, delete or anonymize it to ensure that the personal information involved cannot be recovered.

e) If a model deployed on multiple devices or through cloud-edge collaboration is decommissioned and removed, destroy the model and data from the multiple devices and the cloud at the same time.

7 Confirmation Methods

7.1 General

The confirmation method for general machine learning algorithm security requirements is as follows:

a) Determine whether it has valid certification materials, proving that the organization or individual possesses Level 2 qualifications as specified in GB/T 37988-2019.

b) The confirmation methods corresponding to machine learning algorithm confidentiality requirements are as follows:

1）In the verification and testing stage, check that the algorithm has passed tests on the prevention of membership inference attacks and inversion attacks.

2）Perform recovery testing on destroyed data and models to determine whether they can be recovered;

3）Check whether it has set access permissions for algorithms, models, and application products.

c) The confirmation methods corresponding to machine learning algorithm integrity requirements are as follows:

1）In the algorithm design, development, and application deployment stages, check whether regular verification of the data, model, and runtime environment has been performed.

2）Check whether tests have been conducted on indicators such as dataset size, balance, accuracy, and relevance to algorithm tasks and determine whether the test results show that the algorithm requirements proposed in the design documentation are met.

d) The confirmation methods corresponding to machine learning algorithm availability (可用性) requirements are as follows:

1）Check whether publicly disclosed vulnerabilities have been fixed.
2）Check whether restrictions for attributes such as data formats and size are included in the design documentation.
3）Check whether tests have been conducted on the time it takes for the algorithm to return to the operating state and degree of recovery after an algorithm failure event occurs.
4）Check whether tests have been conducted on the accuracy of the data, including but not limited to sampling testing and full testing.

e) The confirmation methods corresponding to machine learning algorithm controllability requirements are as follows:

1）Check whether test data such as the number of algorithm failures, the severity and frequency of failures, and the proportion of algorithm failures are recorded in the test documentation, and confirm whether the test results meet the threshold values.
2）Check whether indicators such as the algorithm's normal service time, cumulative effective service time, and normal service time proportion have been recorded in the deployment and operation stage.
3）Check whether documentation materials and log records of the algorithm's key decision-making points have been compiled.

f) The confirmation methods corresponding to machine learning algorithm robustness requirements are as follows:

1）Check whether lossy data such as compressed and corrupt data and interference data such as data with added noise or data that have been transformed (变换) have been used to test the correctness of algorithm functions.
2）Check test records or test reports issued by third parties to determine whether the accuracy in white-box and black-box adversarial scenario tests meets the set threshold.
3）Check test records or test reports issued by third parties to determine whether the success rates in adversarial example attack, physical adversarial attack, and backdoor attack tests meet the set thresholds.
4）Check the design documentation to determine whether thresholds for feedback output, query times, and query frequency have been set.

g) The confirmation methods corresponding to machine learning algorithm privacy requirements are as follows:

1）Check whether the personal information authorization records are consistent with the data scale, and spot check whether the personal information and corresponding authorization records are complete.

2）Check whether data masking measures are adopted for personal information during data processing.

7.2 Design and Development Stage

In the design and development stage, the confirmation methods for machine learning algorithm security requirements are as follows:

a) Check the contaminated data detection records, determine whether corresponding contaminated data detection and repair methods and repair results are provided, and test whether the A.2 Integrity Indicators training data consistency and test data consistency test sub-items meet the design requirements.

b) Interview the person in charge of machine learning algorithm design and development, ask about the availability requirements of the algorithm, determine whether there is an availability analysis document for the algorithm, and test whether A.3 Availability meets the design requirements.

c) Confirmation method for judging that the training data meets indicator (指标) requirements:

1）Interview the person in charge of dataset and machine learning algorithm design and development, check the training dataset records and documentation, test the A.2 Integrity Indicators data accuracy test sub-item, and check whether the dataset scale meets the design requirements.

2）Interview the person in charge of dataset and machine learning algorithm design and development, check the training dataset records and documentation, test the A.2 Integrity Indicators data balance test sub-item, check the balance indicator and balance indicator rationality analysis sections in the design documentation, and check the datasets or data statistical materials to determine whether they meet the design requirements.

3）Test the A.2 Integrity Indicators data accuracy test sub-item, check whether the data labeling accuracy meets the design requirements, and check whether data verification and recording have been implemented during the algorithm training process.

d) Interview the person in charge of machine learning algorithm emergency management and check system documents and design documentation to see whether an emergency response mechanism for accidents has been set up, whether an accident handling process has been clearly spelled out, and whether an accident information postmortem review mechanism has been

set up.

e) **Interview the person in charge of machine learning algorithm design and development, ask whether the algorithm is used for decision-making related to key matters such as life and property safety and protection of individual rights in important fields such as public services, transportation and driving, financial services, health and hygiene, and welfare and education, and check the robustness assessment records to determine whether A.5 anti-attack capabilities meets the design requirements.**

7.3 Verification and Testing Stage

In the verification and testing stage, the confirmation methods for machine learning algorithm security requirements are as follows:

a) Test the A.5 Robustness Indicators sub-items including digital world white-box adversarial robust accuracy (对抗鲁棒准确率), digital world black-box query attack adversarial robust accuracy, digital world transfer-based attack (迁移攻击) adversarial robust accuracy, physical world adversarial example attack success rate, and model backdoor attack success rate to check whether the above indicators meet the design requirements.

b) If a third party is entrusted with testing, check whether the test documents contain records on algorithm and data confidentiality security mechanisms.

c) Test the A.2 Integrity Indicators data repeatability test sub-item to check whether the repeatability of the test dataset and the training dataset meets the design requirements.

d) Reproducibility testing confirmation method:

1）Inspect the test documentation to determine if it includes setup reproducibility (设置可复现性) thresholds and maintains the analysis process.

2）Test the A.3 Availability Indicators accessibility test sub-item and test A.4 Controllability Indicators lossy data robustness, interference data robustness, and other test sub-items, check the accessibility indicators and controllability indicators in the assessment index system and corresponding indicator measurement descriptions, and check whether the same data can reproduce the same or highly similar results when the test dataset is used to verify the model.

e) **Confirmation method for judging that the test data meets indicator requirements:**

1）Test the A.2 Integrity Indicators data scale test sub-item, check the

data scale indicator and data scale indicator rationality analysis sections in the design documentation, and check the test datasets or data statistical materials to determine whether the test data scale meets the design requirements.

2）Test the A.2 Integrity Indicators data balance test sub-item, check the balance indicator and balance indicator rationality analysis sections in the design documentation, and check the datasets or data statistical materials to determine whether the test data balance meets the design requirements.

3）Test the A.2 Integrity Indicators data accuracy test sub-item, check whether the data labeling accuracy meets the preset threshold, and check whether data verification and recording have been implemented during the algorithm training process.

4）Test the A.2 Integrity Indicators data task correlation test sub-item, check the data task correlation indicator and data task correlation indicator rationality analysis sections in the design documentation, and check the test datasets or data statistical materials to determine whether the test data correlation meets the design requirements.

f) Test the A.5 Robustness Indicators sub-items including digital world white-box adversarial robust accuracy, digital world black-box query attack adversarial robust accuracy, digital world transfer-based attack adversarial robust accuracy, physical world adversarial example attack success rate, and model backdoor attack success rate to check whether the above indicators meet the design requirements.

g) Check the test documentation to determine whether it includes methods or measures to detect manual algorithm takeover and termination and check the rationality of simulated scenarios of improper algorithm behavior.

7.4 Deployment and Operation Stage

In the deployment and operation stage, the confirmation methods for machine learning algorithm security requirements are as follows:

a) Interview the person in charge of the machine learning algorithm, check the design and deployment documents, test the A.3 Availability Indicators data quality test sub-item, and check whether input data requirement tests were performed. Test the A.5 Robustness Indicators sub-items including lossy data robustness, interference data robustness, digital world white-box adversarial robust accuracy, digital world black-box query attack adversarial robust accuracy, digital world transfer-based attack adversarial robust accuracy,

physical world adversarial example attack success rate, and model backdoor attack success rate to check whether the above indicators meet the design requirements.

b) Interview the person in charge of deployment and operation, check the design documentation and deployment and operation documentation, test the A.1 Confidentiality system information confidentiality indicator sub-items, test whether the A.1 Confidentiality data confidentiality indicator and operation data confidentiality indicator sub-items meet the design requirements, test the A.5 Robustness anti-attack capability indicators test sub-items, and check whether access control testing has been carried out.

c) Test the A.1 Confidentiality Indicators model confidentiality test sub-item and verify whether methods such as model encryption and code obfuscation have been adopted to prevent the inversion (逆向) of model parameter files and code files.

d) Interview the person in charge of deployment and operations, check the relevant design and operation documentation, test the A.3 Controllability Indicators auditability documentation completeness test sub-item, and check whether the documentation completeness test has been carried out.

e) Interview the person in charge of the machine learning algorithm and check whether the design and operation documentation records the deployment of data security protection mechanisms, including but not limited to data encryption algorithms and integrity checks.

f) Interview the persons in charge of the machine learning algorithm and emergency management, test the A.3 Availability recoverability sub-item, check the design documentation and runtime environment emergency response mechanism, check the deployment of emergency response mechanisms for accidents in the algorithm system, product, or service, and check whether the accident handling process is clearly stated to ensure manual emergency intervention and termination capabilities when the machine learning algorithm causes security problems.

g) Interview the person in charge of the machine learning algorithm, review the design and deployment documentation, test the A.3 Controllability Indicator auditability documentation completeness test sub-item, and verify whether the model decision-making or computation process is transparent, explainable (可解释性), and auditable. If it is unexplainable, determine whether the algorithmic decision results are used only as an auxiliary decision-making support for important decisions.

7.5 Maintenance and Upgrading Stage

In the maintenance and upgrading stage, the confirmation methods for machine learning algorithm security requirements are as follows:

a) Check the algorithm change record file, test whether the A.1 Confidentiality Indicators model confidentiality test sub-item meets the design requirements, and test whether A.3 Availability Indicator sub-items algorithm accuracy, software and hardware dependencies, computing resources availability, recovery time, recoverability, and accessibility meet the design requirements.

b) Check the model upgrade verification record file, test the A.2 Integrity Indicators model consistency test sub-item, determine whether the actually deployed model is the same as the pre-deployed model, and determine whether security verification mechanisms for model upgrade package files have been set up and verified.

7.6 Decommissioning and Removal Stage

In the decommissioning and removal stage, the confirmation methods for machine learning algorithm security requirements are as follows:

a) Check the algorithm-related design and operations documentation to see whether it sets conditions to be met for decommissioning and removal and sets a reasonable time period for the destruction of data and models.

b) Check the data storage medium to see whether the data that should be deleted is destroyed and check whether the retained data has been obfuscated and encrypted. Test the A.1 Confidentiality Indicators data confidentiality indicators, use data recovery means to check data storage media, and confirm whether the destroyed data in each device can be recovered.

c) Check whether the algorithm has been destroyed in the storage medium, test the A.1 Confidentiality Indicators model confidentiality indicators, use data recovery means to check the algorithm storage and operation medium, and confirm whether the destroyed algorithm in each device can be restored.

d) Test the A.6 Privacy Indicators data compliance and personal information protection test sub-items to detect whether the personal information protection test has been carried out.

8 Security Assessment Implementation

8.1 Security Assessment Format

Machine learning algorithm security assessment is divided into two formats: self-assessment and third-party assessment. Self-assessment and third-party assessment are combined and complement each other.

Self-assessment refers to a security assessment of an organization's machine learning algorithms initiated by machine learning algorithm developers and operators. Usually, self-assessment is initiated by the organization's internal quality management department or testing department. Self-assessment, mainly in the form of inspection and testing, shall be implemented under the guidance of this Standard and in combination with algorithm-specific security requirements. Periodically conducted self-assessments can be appropriately simplified in the assessment process. In order to ensure security assessment implementation, relevant parties involved in and that influence the security of the algorithm shall also give their cooperation.

Third-party assessment can implement a complete machine learning algorithm assessment process according to the requirements of this Standard. Third-party assessments can also assess algorithms or related content on the basis provided by the implementation of self-assessments. Third-party assessment is carried out by professional assessment agencies of relevant parties such as independent machine learning system developers and operators and mainly takes the form of a combination of interviews, inspections, and tests.

8.2 Security Assessment Preparation

8.2.1 Clarify Assessment Scope

Select the algorithm for which assessment is to be implemented - model documentation and other explanatory documentation that influences machine learning security, algorithm operation (prototype) code, data, and deployment environment information.

Data includes training data and production data, with production data determined with a view to the assessment of the operating situation of the operating entity and the permissions of the data subject. The deployment environment includes the network boundaries, application software, and computing and storage resources that carry the algorithm.

8.2.2 Start of Assessment Work

Before the assessment work is carried out, the background, objectives, principles, and basis of the machine learning algorithm security assessment shall be clarified, the

industry and field standards and policy documents of the object to be assessed shall be fully investigated, an assessment team shall be established, and assessment tasks shall be determined. In third-party assessment, confidentiality agreements shall be signed with application developers and operation managers.

8.2.3 Assessment Scheme Formulation

Taking the specific situation of the object of evaluation into account, an assessment scheme shall be prepared, which shall include:

a) Assessment scope, objects, objectives, etc.;

b) Assessment content, implementation methods, time and progress schedule, and the software and hardware tools and environments used, such as the test sets and adversarial example sets determined according to the calculation volume, assessment time, and environment used by the model;

c) Risk control measures;

d) Personnel arrangement and project management system;

e) List of matters requiring the cooperation of the assessed party;

f) The documents, code, and other relevant materials required for different stages of assessment to be prepared by the assessed party.

8.2.4 Assessment Scheme Review

Evaluate the feasibility, applicability, and pertinence of the formulated assessment scheme.

8.3 Security Assessment Execution

Conduct item-by-item assessment according to the confirmation methods put forward in Chapter 7 of this document.

8.4 Security Assessment Summary

8.4.1 Comprehensive Analysis

When assessing each life cycle of a machine learning algorithm, the evaluator shall analyze and judge the security issues and potential risks discovered during the assessment process and conduct a comprehensive analysis and assessment on the security of the machine learning algorithm. Comprehensive analysis shall mainly include:

a) Compliance with the indicators in the confidentiality, integrity, availability, controllability, robustness, and privacy units in this Standard;

b) Non-applicable items in Part 6 of this Standard and explanations;

c) Carry out risk analysis on some of the risk issues that meet or fail to meet

indicator items;

d) Assess other security issues discovered;

e) Relevant suggestions for some compliance or non-compliance indicators and other security issues.

8.4.2 Report Preparation

The evaluator shall produce an assessment report on the results of this assessment, including but not limited to a description of the object of assessment, the assessment time, and a description as to whether the algorithm meets the assessment requirements.

8.4.3 Result Feedback

The evaluator shall provide feedback on the assessment work situation and the formal assessment report to the assessed party.

8.5 Security Assessment Result Judgment

Result judgments on the machine learning algorithm security assessment include the following three categories:

a) If there are unsatisfied items in the above basic-level requirements, the result is judged to be "does not meet the basic-level requirements of this Standard."

b) If it meets all the basic-level requirements, it is judged to be "meets the basic-level requirements of this Standard."

c) If it meets all the basic-level and enhanced-level requirements, it is judged to be "meets the enhanced-level requirements of this Standard."

Appendix A
(Normative)
Machine Learning Algorithm Security Assessment Index System

Appendix A gives a fine-grained assessment index system for machine learning algorithm security, which can be used to carry out relevant assessment activities on machine learning algorithms in different application scenarios.

This index system consists of two layers of "attributes-indicators," in which the attributes include confidentiality, integrity, availability, controllability, robustness, and privacy. The assessment index system is composed of the assessment indicators corresponding to each attribute. The assessment indicator descriptions and list of subdivided items are given below.

During the assessment process, corresponding indicators shall be selected according to the maturity of different machine learning technologies and the security requirements of different application fields. Ensure that the application objectives and usage methods comply with national laws and regulations, industry regulatory policies, standards, specifications, and ethical requirements.

A.1 Confidentiality Indicators

Confidentiality indicators are used to assess the confidentiality of each stage of the machine learning algorithm life cycle. They include but are not limited to:

a) Data confidentiality indicators: Assess the risk of unauthorized persons using gradient information, machine learning system output, and other information to steal machine learning-related data through reverse engineering and other technical means. These indicators include training data confidentiality, operating data confidentiality, and the ability to resist membership inference attacks, involving the confidentiality of data transmission, storage, computation, and aggregation as well as personal information protection and key security.

b) Model confidentiality indicators: Assess the confidentiality of the model, infer the parameters or functions of the machine learning model through continuous access to the system being tested, and judge the similarity between the stolen or inferred training data and the original data.

c) System information confidentiality indicators: Assess the theft of or unauthorized access to software and hardware dependency information at each stage in the machine learning algorithm life cycle.

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
1	Data confidentiality indicators	Training data confidentiality	The similarity between the stolen training data obtained through gradient information, membership inference, or another method and the original data	Draw a curve with the number of queries as the x-axis and the similarity deviation ratio between the imitation (仿制) training data and the imitated (被仿制) training data as the y-axis, calculated using 1-similarity (1-相似度计算)	The greater the similarity and the higher the attack success rate, the lower the security	Required
2		Membership inference attack success rate	For a set of test examples, infer the proportion that belongs to the training dataset	For a set of test examples, infer the proportion that belongs to the training dataset	The lower the proportion, the better the security	Required
3		Operating data confidentiality	The similarity between the stolen training data obtained through machine learning system output information and the original data	Draw a curve with the number of queries as the x-axis and the similarity deviation ratio between the imitation training data and the imitated training data as the y-axis, calculated using 1-similarity	The greater the similarity and the higher the attack success rate, the lower the security	Required
		Data destruction	Whether data that should have been deleted were not destroyed or can still be recovered	Detect whether data that should have been deleted can be obtained directly or through recovery techniques	If the data cannot be obtained, the security is better; if it can be obtained, the security is worse	Required
4	Model confidentiality indicators	Model stealing success rate	The similarity and performance of the stolen machine learning model relative to the original model	Draw a curve with the number of queries as the x-axis and the deviation ratio between the imitation model recognition performance and the imitated model recognition performance as the y-axis	The smaller the area under the curve, the better the security	Required
		Model destruction	Whether algorithms that should have been deleted were not destroyed or can still be recovered	Detect whether algorithms that should have been deleted can be obtained directly or through recovery	If the algorithms cannot be obtained, the security is better; if they can be	Required

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
				techniques	obtained, the security is worse	
		Model anti-inversion	Whether it is possible to obtain the model parameter files and code files through inversion	Detect whether technical means such as model encryption and code obfuscation are applied to prevent model inversion	The more comprehensive the use of technical means, the better the security	Required
5	System information confidentiality indicators	System information confidentiality	The degree to which algorithm, data, dependency, and other information used by the system can be accessed by unauthorized entities	Detect the number of times the algorithm, data, dependency, and other information used by the system can be accessed by unauthorized entities	The less information that can be obtained, the better the security	Required
6	Other indicators	Other relevant indicators	Degree to which other relevant indicators meet confidentiality objectives	Determined based on indicators	Determined based on test indicators	Optional

A.2 Integrity Indicators

Used to assess the integrity of each stage of the machine learning algorithm life cycle. They include but are not limited to:

a) Data integrity indicators: Assess the accuracy and reliability of the data involved in the algorithm, in terms of the integrity, clarity, accuracy, and reliability of the data in each stage of the machine learning life cycle. These indicators include data consistency, data balance, and data accuracy.

b) Runtime environment integrity: Assesses the integrity level of the runtime environment on which the algorithm depends, including the consistency of runtime environment information.

c) Model integrity indicators: Assess the degree of impact of unauthorized replacement or destruction of the SDK or product carrying the algorithm and measure whether the deployed model is consistent with the original model.

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
1	Data integrity indicators	Data balance	Including the degree of data category balance and unbiasedness	Use statistical methods to detect the data category distribution and bias	If the data categories are balanced and there is no bias,	Required

					the data balance is good	
2		Data accuracy	The degree to which the data describes the true values of the actual object	Use a method combining sampling tests and full tests to check whether the data can accurately represent the actual objects it describes	The higher the correctness ratio in the detection sample, the better the data accuracy	Required
3		Training data consistency	Measure the degree of consistency between the falsified (篡改) training data and the original training data	Use statistical means to calculate the [consistency] ratio of the falsified training data to the original training data	The higher the consistency, the better the security	Required
4		Testing data consistency	Measure the degree of consistency between the falsified test data and the original test data	Use statistical means to calculate the [consistency] ratio of the falsified test data to the original test[1] data	The higher the consistency, the better the security	Required
5		Data repeatability	Measure the degree of repeatability between test data and training data	Use statistical means to calculate the repetition ratio between test data and training data	The lower the data repetition, the better the test data repeatability	Required
6		Data task correlation	Measure the task correlation of test data	Use statistical means to calculate the degree of correlation between the test data and training data distribution	The higher the data task correlation, the better the security	Required
7		Training data scale	Measure the degree of abundance of training data	Use statistical means to calculate the training data sample size	The larger the training data scale, the better the security	
8		Test data scale	Measure the degree of abundance of test data	Use statistical means to calculate the test data sample size	The larger the test data scale, the more accurate the test	

[1] Translator's note: The Chinese source text has "training data" (训练数据) here, which is presumably a typo that should be corrected to "test data" (测试数据).

9		Data labeling accuracy	Check the accuracy of data labeling	Use statistical methods to detect the ratio of correctly labeled data to all data	The higher the labeling accuracy, the better the security	
10	Model integrity indicators	Model consistency	Measure whether the model is consistent	Check whether the deployed model is consistent with the original model	If they are consistent, the security is better	Required
11	Runtime environment integrity indicators	Runtime environment information consistency	Measure the consistency of system information such as runtime dependencies after falsification relative to the original information	Use code unit testing, module component testing, integration testing, and other methods to test the consistency of the various runtime environments on which the algorithm depends	The higher the consistency, the better the security	Required
12	Other indicators	Other relevant indicators	Compliance of other relevant indicators with integrity objectives	Determined based on indicators	Determined based on test indicators	Optional

A.3 Availability Indicators

Used to assess the availability of each stage of the machine learning algorithm life cycle. They include but are not limited to:

a) Usability indicators: Assess the usability of algorithms, the level of dependence of software and hardware environments, and the usability of computing resources within the effective life cycle. These indicators include algorithm accuracy, accessibility, computing resource availability, and software and hardware dependency.

b) Recoverability indicators: Assess the time and investment required to restore algorithms and data from a disaster state to an operational state. These indicators include recovery time and recoverability.

c) Training data quality indicators: Assess the accuracy of the training data in representing the actual objects it describes.

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
1	Usability indicators	Algorithm accuracy	Includes indicators based on accuracy, precision, recall,	Select indicators according to the algorithm characteristics, such as	The higher the value of each indicator, the	Required

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
			F1-score, PR curve, and AUC	accuracy, precision, recall, F1-score, PR curve, and AUC to assess the usability of the algorithm.	stronger the algorithm availability	
2		Software and hardware dependency	The degree of availability impact of attacks on the software and hardware required in the machine learning life cycle	Based on publicly disclosed vulnerabilities, use penetration attacks and other attack methods to interfere with the normal operation of the software and hardware upon which algorithm operation depends	The lower the dependence on software and hardware, the higher the security	Required
3		Computing resource availability	Assess the availability impact of attacks on computing resources such as CPU and GPU	Use attack methods targeting CPU, GPU, and other computing resources to calculate the algorithm operating conditions under different attacks	The stronger the ability to resist interference, the higher the security	Required
4		Accessibility	Detect the ability to obtain the operation results of the algorithm when it needs to be accessed within a certain period of time	Use the sampling method to calculate the effective algorithm access quantity and proportion within a period of time	The higher the accessibility proportion under a given effective quantity, the better the security	Required
5		Input data requirements	Including whether restrictions for attributes such as data formats and size are added	Conduct tests using data with different attributes such as format and size to determine whether there is restriction logic	The more restrictions on input data attributes, the better the security	Required
6	Recoverability indicators	Recovery time	After a security incident occurs, the average time between the instant when the algorithm is disabled causing a suspension of services and the instant when the algorithm is	Under given time and conditions, simulate security incidents and calculate the average, maximum, mode, and median of the algorithm recovery time	The lower the recovery time statistical indicators, the higher the security	Required

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
			restored to support the operation of the intelligent system			
7		Recoverability	After a security incident, the effective degree of recovery from the failure of the algorithm to the availability of the fix	Under given time and conditions, simulate security incidents and calculate the proportion of full function recovery	The higher the proportion, the higher the security	Required
8	Data quality indicators	Data quality	Includes accuracy (the degree to which the data accurately represents the true values of the actual objects it describes) and specification compliance (the degree to which the data complies with data standards, data models, business rules, metadata, or authoritative reference data)	Use a method combining sampling tests and full tests to check whether the data can accurately represent the actual objects it describes	The higher the correctness ratio in the detection sample, the better the data accuracy	Required
9	Other indicators	Other relevant indicators	Compliance of other relevant indicators with availability objectives	Determined based on indicators	Determined based on test indicators	Optional

A.4 Controllability Indicators

Used to assess the controllability of each stage of the machine learning algorithm life cycle. They include but are not limited to:

a) Durability indicators: Assess the severity and frequency of failures during continuous operation under specified times and conditions, including the algorithm's continuous operating status. (Enhanced: Emergency response mechanisms for accidents, including manual emergency intervention

mechanisms, shall be set up in the system, product, or service; the accident handling process shall be clarified to ensure a prompt response when an artificial intelligence (AI) security risk occurs, such as stopping the production of problematic products and recalling problematic products; accident information postmortem review mechanisms have been set up).

b) Auditability indicators: Assess the completeness of management process documentation and the traceability and auditability of key links, including the completeness of documentation,

c) Sustainability indicators: Assess the ability of the algorithm to operate normally, including normal service time.

d) Fault tolerance indicators: Assess the algorithm's ability to avoid algorithm failure and its fault tolerance, including failure density and failure resolution rate.

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
1	Durability indicators	Continuous operation	The severity, frequency, and fault tolerance of failures that occur during continuous operation under specified conditions and during a specified period of time	Under the specified time and conditions, calculate statistics on the severity and frequency of algorithm failures and calculate the fault-tolerance policy boundaries according to fault-tolerant test examples	The longer the continuous operation time, the better the security	Required
2	Audit-ability indicators	Documentation completeness	There are complete documentation materials as well as log records of the algorithm's key decision-making points	Based on the requirements of each stage of project management, documentation materials are complete; assess their auditability and traceability	The more complete the materials, the higher the security	Required
3		Cumulative effective service time	The sum of effective service time provided	Calculate the effective service duration for the specified test period	Within the specified time, the longer the service duration, the higher the security	Required
4		Normal service time	The time during which the algorithm can be normally used during a specified period of time under specified conditions	Calculate the algorithm's normal service time under the specified time and conditions	The longer the normal service time, the higher the security	Required
5		Normal service	Assess the machine	Under the specified time	The higher the	Required

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
		time proportion	learning system's normal service time period as a proportion of the total service time period	and conditions, based on the input data of the test dataset, calculate the normal service time period as a proportion of the total service time period	proportion, the better the security	
6	Fault tolerance indicators	Failure density	Within a certain test period, calculate the actual performance status of the algorithm; incorrect prompt information and failure of the model to perform as expected are counted as failures	Calculate the number of algorithm failures within a certain test period	The lower the number of failures, the better the security	Required
7		Failure resolution rate	The ratio of detected failures to resolved failures	After calculating the actual performance of the algorithm within a certain test period, this is the proportion of resolved issues to detected security issues	When the number of failures is not zero, the higher the ratio, the better the security; when the number of failures is zero, the result of this item is not calculated	Required
8		Other relevant indicators	Compliance of other relevant indicators with controllability objectives	Determined based on indicators	Determined based on test indicators	Optional

A.5 Robustness Indicators

Used to assess the robustness of each stage of the machine learning algorithm life cycle. They include but are not limited to:

a) Correctness indicators: Assess the ability of the algorithm to operate normally when confronted with abnormal data inputs, including resistance to interference data and lossy data.

b) Anti-attack capability indicators: Assess the ability of the algorithm to operate normally when confronted with attacks, including the ability to resist digital world attacks, physical world attacks, black- and white-box attacks, and backdoor attacks.

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
1	Correct-ness indicators	Lossy data robustness	Calculate the algorithm's functional implementation correctness based on test results	Use lossy data such as compressed and corrupt data to perform testing, and calculate the algorithm's functional implementation correctness based on test results	The higher the correctness of algorithm function implementation, the better the robustness	Required
2		Interference data robustness	Calculate the algorithm's functional implementation correctness based on test results	Use interference data such as data with added noise or that has been transformed to perform testing, and calculate the algorithm's functional implementation correctness based on test results	The higher the correctness of algorithm function implementation, the better the robustness	Required
3	Anti-attack capability indicators	Digital world white-box adversarial robust accuracy	Calculate the accuracy of algorithm recognition in white-box adversarial scenarios	Draw multiple curves with the perturbation (扰动) size of the generative adversarial examples and the number of attack iteration rounds as the x-axis, and the recognition accuracy of the corresponding model as the y-axis	Select some points along the x-axis, calculate the value of the curve at these points, and take the average. The higher the average value, the better the security	Required
4		Digital world black-box query attack adversarial robust accuracy	Calculate the accuracy of algorithm recognition in black-box adversarial scenarios	Draw multiple curves with the perturbation size of the generative adversarial examples and the number of queries as the x-axis, and the recognition accuracy of the corresponding model as the y-axis	Select some points along the x-axis, calculate the value of the curve at these points, and take the average. The higher the average value, the better the security	Required
5		Digital world transfer-based attack adversarial robust accuracy	Calculate the accuracy of algorithm recognition in transfer-based attack scenarios	Draw multiple curves with the perturbation size of the generative adversarial examples as the x-axis, and the recognition accuracy of the corresponding model as the y-axis	Select some points along the x-axis, calculate the value of the curve at these points, and take the average. The higher the average value, the	Required

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
					better the security	
6		Physical world adversarial example attack success rate	Calculate the proportion of successful attacks in physical adversarial scenarios	Generate physical world adversarial examples and calculate the attack success rate based on the attack results	The lower the success rate, the better the security	Required
7		Model backdoor attack success rate	Calculate the proportion of successful model backdoor attacks	Try to implant a backdoor and superimpose triggers (触发器) on input examples, and then calculate the attack success rate based on the attack results	The lower the success rate, the better the security	Required
8		Access controls	Whether restriction measures are adopted to control the information attackers can obtain	Detect whether access control measures such as limiting the feedback output of the model, limiting the number of queries of the model, and limiting the usage frequency of accounts and IPs are adopted		
9	Other indicators	Other relevant indicators	Compliance of other relevant indicators with robustness objectives	Determined based on indicators	Determined based on test indicators	Optional

A.6 Privacy Indicators

Used to assess the privacy of each stage of the machine learning algorithm life cycle. They include but are not limited to:

a) Compliance indicators: Assess the degree of compliance at each stage of the data life cycle, including data compliance.

b) Protection indicators: Assess the level of data privacy leak prevention and security protection at each stage of the machine learning life cycle, including data leakage prevention, banking, and personal information protection.

No.	Indicator category	Test sub-item	Test sub-item detection target	Test method	Measured value description	Type
1	Compli-ance indicators	Data compliance	The degree of legal compliance in data collection and other	Conduct assessment in accordance with existing personal information	The higher the legal compliance, the better the	Required

			stages	protection and data security policies, regulations, and standards	security	
2	Protection indicators	Leak prevention	The similarity of private data restored based on the gradient information and the original private data	Use the gradient information attack method and calculate the similarity and proportion between the restored private data and the original data	The lower the similarity and proportion, the better the security	Required
3		Data concealment (隐含性)	Assess the likelihood that the machine learning model will remember or reveal training data	Use inversion methods and calculate the proportion of data inferred from the machine learning model	The lower the proportion, the better the security	Required
4		Personal information protection	Assess whether protective measures are taken for personal information at each stage of the algorithm [life cycle]	Use technologies, documents, and on-site inspections to check whether personal information protection methods and measures such as homomorphic encryption and transmission channel encryption have been adopted for key scenarios	The more masked data and the less dirty data, the better the security	Required
5	Other indicators	Other relevant indicators	Compliance of other relevant indicators with availability objectives	Determined based on indicators	Determined based on test indicators	Optional

A.7 Indicator Calculation Methods

Methods for calculating machine learning algorithm security assessment indicators include the following three categories:

a) Expert assessment: For the calculation of indicators such as auditability and rationality, industry experts can be organized to conduct comprehensive assessment.

b) Dataset calculation: For unexpected failures in availability and robustness and other indicators, use the test dataset for calculation.

c) Simulated attack calculation: For indicators of intentionality (有意动机) in confidentiality and robustness, use simulated security attack methods for on-site calculation.

A.8 Indicator Calculation and Publication Requirements

The calculation of security assessment indicators for machine learning algorithms depends on the selection and construction of security attack methods and security test datasets. In order to ensure the transparency and impartiality of security assessment indicators, the calculation and publication of machine learning algorithm security assessment indicators shall comply with the following requirements:

a) For security assessment indicators calculated by measuring the performance of machine learning algorithms under security attacks, various types of security attack methods with relatively good attack performance at the current time shall be selected to perform comprehensive assessment, and the security attack methods and key parameter settings used shall be clarified and made public during calculation and publication.

b) For the security assessment indicators calculated by measuring the performance of machine learning algorithms on test datasets, the scale of the test dataset used, the types of test data, typical test data examples, and other key information shall be clarified and made public during calculation and publication.

Appendix B
(for Reference)
Machine Learning Algorithm Security Risks

B.1 Machine Learning Algorithm Classification

A machine learning algorithm is a set of methods by which computers output and improve predictions or behaviors based on data. Based on differences in the information contained in the training examples and their feedback methods, training methods, and construction methods, machine learning algorithms can be classified in different ways.

Based on differences in information contained in the training examples and feedback methods, machine learning algorithms are divided into three categories: supervised learning, unsupervised learning, and reinforcement learning. Among these, supervised learning can be divided into regression algorithms and classification algorithms based on differences in processing tasks; unsupervised learning can be divided into clustering, dimensionality reduction, and association analysis based on differences in processing tasks; and reinforcement learning can be divided into model-based learning and model-free learning based on differences in learning methods.

Based on differences in training methods, machine learning algorithms can be divided into batch learning and online learning.

Based on differences in algorithm construction methods, machine learning algorithms can generally be divided into two types: algorithms obtained by retraining existing third-party algorithms and algorithms obtained by original design and training.

Based on differences in learning principles, machine learning algorithms can generally be divided into three categories: symbolic learning, connectionist learning (联接主义学习), and statistical learning. Typical representatives of symbolic learning algorithms include decision trees and rule-based learning. Connectionist learning algorithms mainly refer to shallow neural networks such as perceptrons and BP networks as well as deep neural networks. Statistical learning algorithms mainly include support vector machines, Bayesian classification, and principal component analysis.

Based on differences in algorithm training methods, machine learning algorithms can generally be divided into online training and offline training.

Based on differences in security hazards caused by the improper application of algorithms to the digital world, the physical world, and human society, machine learning algorithms can generally be divided into three categories: machine learning algorithms that only affect the security of the digital world, machine learning algorithms that affect the security of the physical world, and machine learning algorithms that affect the security of human society. Among these, machine learning algorithms that only affect

the security of the digital world include machine learning algorithms applied in fields such as information communication and leisure and entertainment. Machine learning algorithms that affect the security of the physical world include machine learning algorithms applied in fields such as petroleum, chemical engineering, manufacturing, and agriculture. Machine learning algorithms that affect the security of human society include machine learning algorithms applied in fields such as military, finance, medicine, and transportation.

B.2 Machine Learning Algorithm Vulnerabilities and Attack Threats

B.2.1 Machine Learning Algorithm Vulnerabilities

Factors such as the limitations of machine learning technology, machine learning algorithm design errors, machine learning algorithm software defects, lack of or lax data security management, and security vulnerabilities in machine learning frameworks all lead to security vulnerabilities for machine learning algorithms. These include:

a) Limitations of machine learning technology: Machine learning algorithms have technical limitations that have not yet been overcome, such as weak robustness, unexplainability, and bias and discrimination. Weak robustness means that machine learning algorithms may produce unexpected and incorrect results when faced with actual application scenarios that are complex and changeable, involve abnormal malicious interference, and other such situations. Unexplainability means that humans cannot understand the internal operating logic, the meaning of parameters obtained through training, and the reasons for decisions of machine learning algorithms based on deep neural networks. This poses challenges to problem location, debugging and modification, and accountability investigation for machine learning algorithms. Bias and discrimination mean that machine learning algorithms learn the solutions to problems autonomously from training datasets that contain existing social discrimination, which will produce latent biases in decision-making results.

b) Machine learning algorithm design errors: The design of machine learning algorithm objective functions, solution methods, or other features are incorrect and the algorithm cannot achieve the designer's preset goals. This results in unpredictable and uncontrollable behaviors that deviate from expectations.

c) Machine learning algorithm software defects: Machine learning algorithms have not implemented effective software quality management in the design, R&D, maintenance, and upgrading stages, resulting in security vulnerabilities.

d) Lack of or lax data security management: During the full machine learning algorithm life cycle, management during stages such as the training of

machine learning algorithms and the collection, storage, use, and entrusted processing, sharing, transfer, and public disclosure of data is not strictly in accordance with national laws and regulations or standards and specifications, which produces many types of risks such as user privacy leaks and non-standardized data preprocessing.

e) Security vulnerabilities in machine learning frameworks: The machine learning framework and the third-party libraries it relies on have security vulnerabilities and backdoors, which produce corresponding security risks for the machine learning algorithms that are developed based on them.

B.2.2 Machine Learning Algorithm Attack Threats

New security attack threats that target machine learning algorithms include adversarial example attacks, data poisoning attacks, algorithm backdoor attacks, model stealing attacks, and private information theft (隐私窃取) attacks:

a) Adversarial example attacks: Machine learning models suffer adversarial example attacks during the running stage. Adversarial examples refer to examples created when the attacker maliciously adds a tiny amount of noise invisible to the human eye to normal examples, causing the model to make an error. Because machine learning models may be deployed in uncontrolled real-world scenarios, adversarial example attacks become a real security risk.

b) Data poisoning attacks: Machine learning models suffer data poisoning attacks during the training stage. Data poisoning refers to when the attacker adds some maliciously constructed examples to the training data so that the trained model has security problems, such as attacks that skew model predictions, reduce model accuracy, and insert backdoors. The forms of data poisoning attacks include directly modifying training data and using feedback to mislead.

c) Algorithm backdoor attacks: An algorithm backdoor attack refers to the insertion of a backdoor into a machine learning model. The models that suffer backdoor attacks perform well in normal data, but produce irrational mispredictions when encountering specific data so as to achieve the attacker's purpose. Methods for inserting backdoors into a model include modifying the training data and modifying the model parameters.

d) Model stealing attack: Machine learning models suffer model stealing attacks during the running stage. The attacker hopes to construct a substitute model by using the information obtained by accessing the model in order to steal specific functions of the attacked model.

e) Private information theft attack: Private information in training data used by the machine learning model is also at risk of theft. The attacker hopes to use

membership inference (judging whether a certain piece of data is in the training dataset), data inversion restoration (restoring the training data), property inference (属性推理; judging whether a feature is used to train the model), and other methods to obtain some information about the training data used by the model. Private information theft attacks can lead to the malicious use of users' sensitive information.

B.3 Security Risks in the Design and Development Stage

The design and development stage is mainly used to clarify tasks, collect and form datasets, design corresponding machine learning algorithms, and select machine learning frameworks. This stage faces security risks at the model, data, and environment levels.

B.3.1 Algorithm-Level Security Risks

When designing machine learning algorithms and models, issues such as correctness need to be considered. Security risks faced at the model level include:

a) Risk that model correctness cannot meet task requirements: The correctness of the algorithm refers to the ability of the algorithm to generate a machine learning model that behaves as expected. Since different tasks have different correctness requirements, when the designed machine learning algorithm cannot guarantee a correctness that meets task requirements, the model will cause security problems during actual application.

b) Risk that model operating efficiency cannot meet task requirements: The model operating efficiency refers to the time spent on algorithm execution. When the designed algorithm requires a long time and cannot meet the task efficiency requirements, such as when the time complexity of the designed algorithm is exponential and cannot meet the needs of a task scenario that requires a response in seconds, such an algorithm will itself become a security risk.

c) Algorithm robustness risk: Because machine learning model training cannot traverse the data of all target fields, the trained model will have poor robustness to external environment interference and malicious attacks. Attackers can exploit this vulnerability of the model to easily construct adversarial examples that produce incorrect model results, resulting in serious security risks.

B.3.2 Data-Level Security Risks

Machine learning mainly conducts training through datasets, learns corresponding

patterns from the datasets, and evaluates the learning performance through various indicators. The scale of datasets, the balance of datasets, the data labeling accuracy, the standardization of data representation, whether data is contaminated, and leaks of private data become security risks in the design and development stage. These include:

a) Risk of insufficient dataset scale: Different machine learning tasks require datasets of different scales. When the dataset scale is not large enough to support the effective learning of machine learning algorithms, it will cause the machine learning algorithms to fail to meet the accuracy requirements corresponding to specific tasks, thus affecting security during model execution.

b) Risk of poor dataset balance: Dataset balance is used to describe the number of examples of different categories contained in a dataset. When the dataset is poorly balanced and some categories have a small number of examples, it will be difficult for the machine learning algorithm to effectively learn the characteristics of the various categories, which will seriously affect the machine learning recognition accuracy for certain categories. In addition, algorithm learning using unbalanced data will lead to model fairness risks.

c) Risk of inaccurate data labeling: For supervised learning algorithms, data labeling is an important constituent part of the dataset and an important factor affecting the security of machine learning. However, most datasets are labeled in the form of manual crowdsourcing. It is difficult to ensure that all the people involved in labeling can label correctly. In cases such as wrong labels or missing labels, it will be difficult for the algorithm to learn correctly, resulting in incorrect classification or prediction problems.

d) Risk of non-standardized data representation: In application scenarios such as intelligent driving, smart finance, and smart health, the original data may contain a large amount of information irrelevant to the task and the data representation may be inappropriate or non-standard. This will cause the machine learning algorithm to fail to accurately learn key features, will affect the accuracy of the algorithm, and may cause a significant drop in algorithm efficiency.

e) Risk of dataset contamination: Datasets collected during the design and development stage may face the risk of malicious data contamination, that is, data poisoning. Due to malicious behaviors that modify the data in the dataset, the machine learning algorithm learns incorrect features, resulting in model errors.

f) Risk of private data leaks due to excessive data collection: When personal information is used without encryption or desensitization, it is easy for attackers to extract data from the model, resulting in the leakage of personal

information in the dataset.

B.3.3 Environment-Level Security Risks

The environment of the machine learning model depends on the framework and third-party libraries it chooses, so vulnerabilities in the framework and third-party libraries themselves will lead to risks such as errors in model operation: Currently, the design and implementation of machine learning algorithms often rely on the support of development frameworks, and these frameworks themselves often use a large number of third-party libraries. Security vulnerabilities and backdoors in the frameworks and third-party libraries used in the design and development stage will lead to corresponding security risks in the machine learning algorithms developed based on them. In addition, the operating system, hardware architecture, and hardware configuration may introduce risks, such as compatibility issues, processing precision issues, security issues, and computing power ("compute") issues.

B.4 Verification and Testing Stage Security Risks

The verification and testing stage verifies and tests the functions and security of the model obtained in the development stage. When the algorithm undergoes data training to obtain a specific model, this phase aims to verify whether the functions and performance of the model meet the pre-defined requirements. In the actual development process, the model reverts back to the design and development stage if it does not meet requirements. This stage faces security risks at the model, data, and environment levels.

B.4.1 Algorithm-Level Security Risks

a) Risk of low model correctness: Different tasks have different correctness requirements. When the correctness of the model on the test dataset cannot meet the requirements, it is necessary to perform redevelopment.

b) Risk of low model operation efficiency: Different tasks have different efficiency requirements. When the efficiency of the model on the test dataset cannot meet the requirements, it is necessary to perform redevelopment and optimize the algorithm.

c) Risk of model private data leakage: The algorithm may expose datasets, and private data can be obtained through simple testing methods.

d) Risk of insufficient algorithm fairness: The algorithm produces output that is biased and violates the principles of fairness.

e) Risk of poor algorithm and model robustness: When the test data contains small deviations or some abnormally distributed data, the results of the

algorithm will be greatly affected, significantly changing the output.

f) Risk of poor model generalizability: When a model is tested using datasets related to actual scenarios and tasks, its generalizability will be poor, making it difficult for the model to output correct results.

B.4.2 Data-Level Security Risks

In the verification and testing stage, a large amount of test data is used to verify the accuracy, efficiency, and robustness of the model. This stage faces security risks such as a high degree of test data and training data repetition, insufficient test datasets, and poor balance. These include:

a) Risk of a high degree of test data and training data repetition: When the data used for testing has a high degree of repetition with respect to the training datasets, the model will perform well on the test data, but its performance on unknown test datasets cannot be verified.

b) Risk of insufficient test dataset scale: If the test data scale is small, it will not be able to effectively verify the various special data that may be encountered, leading the model to make judgment errors on special data.

c) Risk of poor test dataset balance: When the test data balance is insufficient and there is little test data in some categories, it will be difficult to verify the performance of the model on various types of data.

d) Risk of insufficient relevance of test dataset to actual task: When the data related to the actual application scenarios and tasks is not collected for testing, this leads to a problem where the model cannot correctly process actual data during its actual operation process.

B.4.3 Environment-Level Security Risks

Machine learning algorithms are implemented on the basis of a specific framework. In the verification and testing stage, if vulnerabilities in the framework and third-party libraries themselves are not effectively tested, errors will occur during the operation of the model.

B.5 Deployment and Operation Stage Security Risks

The machine learning deployment and operation stage is the process of deploying the machine learning model for actual applications after the model is formed. This stage may be confronted with problems such as unknown actual data, framework updates, attacks on the model, and runtime environment adaptation. This stage mainly faces security risks at the model, data, and environment levels.

B.5.1 Algorithm-Level Security Risks

a) Risk of low model operations efficiency: The efficiency of the machine learning model is tested in the verification and testing stage, but during actual deployment and operation, the efficiency requirements of the real environment may change over time, and models that are not promptly updated may be unable to meet the efficiency requirements of the real environment.

b) Risk of model backdoors: An attacker implants a specific backdoor in the machine learning model so that, although the model's judgment on normal inputs is consistent with the original model, its judgment on special inputs is controlled by the attacker, resulting in specific errors in the model output.

c) Model privacy risks: Attackers can access the model multiple times through the public access interface. Based on the mapping relationship between input and output, the attacker can construct a model that is highly similar to the attacked model without knowing the model parameters and infer and restore the parameter information of the model.

d) Risk of data leakage from data inversion: Security risks from the use of information returned by calls to the model interface to perform an inversion attack in order to restore the training data or some private data.

e) Risk of injection attack: Attackers use security vulnerabilities in the design of the machine learning model to inject malicious commands into the model. When the actual data meets the preset trigger conditions, the model will complete the injected attack behavior.

f) Risk of denial-of-service attack: Attackers exploit model defects or program vulnerabilities to attack models by constructing special data and exploiting sensitive data, causing machine learning service crashes, system memory overflow, or other denial-of-service situations.

g) Explainability risks: Machine learning models are often used in security-sensitive fields such as healthcare, revenue forecasting, and personal information assessment. If the logic of the model output results lacks explainability, it is likely to cause people to question or even object to the validity of the model.

h) Algorithm robustness risks: New algorithms still cannot cover the variable space (可变空间) of the data and cannot make correct judgments for malicious input examples. Attackers can exploit this feature of the algorithm to construct adversarial examples and malicious examples in order to attack the algorithm and cause model errors.

B.5.2 Data-Level Security Risks

After the model is actually deployed, it may encounter risks such as outlier data in the actual environment, data with natural noise perturbation (扰动), special perturbation data, and data contamination. These risks include:

a) Risk of interference data: In the actual environment, there are some extreme input data that are quite different from the rest of the input data. Such outlier data may cause errors in the model output.

b) Natural noise perturbation risk: In the actual environment, normal input data will be affected by environmental factors, and the input may carry unpredictable natural noise perturbations, which lead to deviations in the deployed machine learning model.

c) Risk of special perturbation attack: Alterations to the input using some special method precipitates the machine learning model to output incorrect information. This type of special perturbation attack includes adversarial example attacks.

d) Risk of training data contamination: Some online learning or evolutionary learning machine learning models, after being deployed in the environment, will perform online training based on the actual data and adaptively adjust the model parameters. During this process, the attacker poisons the data so that training models on the contaminated data will gradually cause errors in the model output.

e) Dataset distribution migration (分布迁移) risk: Models usually assume that the training data and real data follow the same distribution, but when the model is deployed in real applications, the dataset distribution may migrate, that is, there is a difference between the real dataset distribution and the training dataset distribution. This results in model output deviations.

f) Risk of data leakage: After a machine learning model is deployed, the attacker may repeatedly call and query the model in order to restore the training data based on the information returned by the model, causing risks such as data leaks.

B.5.3 Environment-Level Security Risks

a) Risk of machine learning framework updates: Most machine learning models are developed based on a specific framework. When the framework and third-party libraries are updated, if corresponding adjustments are not promptly made to the models deployed on this framework, they will have configuration problems and other risks.

b) Software and hardware platform deployment risks: The operating system, hardware architecture, and hardware configuration in the real deployment environment may introduce risks, such as compatibility issues, processing precision issues, security issues, and compute issues.

c) Supply chain risks: In the supply chain process of machine learning models, attackers can reverse crack (逆向破解) machine learning models and inject malicious code through control of software and hardware channels, resulting in the propagation of malicious code.

B.6 Maintenance and Upgrading Stage Security Risks

After a machine learning algorithm or model is deployed, the algorithm and model will be updated and upgraded due to business needs. In this process, when the algorithm and model are updated, it is necessary to consider the corresponding risks in the algorithm development, verification and testing, and deployment and operation stages. When the model upgrade requires data-based retraining, it may face risks such as data contamination. When the framework is updated, algorithms and models need to be promptly upgraded. When the runtime environment, such as software and hardware, changes, corresponding maintenance is also required. This stage mainly faces security risks at the model, data, and environment levels.

B.6.1 Algorithm-Level Security Risks

a) When algorithms and models are significantly adjusted, modified, or upgraded, it is necessary to reconsider the security risks faced by the new algorithms and models in the design and development, verification and testing, and deployment and operation stages.

b) Risk of untimely update of model parameters: When algorithms and models are upgraded, there may be risks such as the model parameters not being updated in time and the incorrect deletion of model parameters.

c) Risk of model configuration conflict: When models are upgraded, the configurations of the new and old models are inconsistent. Failure to upgrade the configuration normally will lead to risks such as the failure of the new model to run normally.

B.6.2 Data-Level Security Risks

During the maintenance and upgrading stage, some machine learning algorithms and models may need to be retrained based on data. In this process, there is a risk of data contamination, that is, attackers may poison the data in order to affect algorithm and model correctness.

a) Data quality risks: During the maintenance and upgrading stage of machine learning algorithms, supplementary data is often added to optimize the current algorithm version. The scale, data representation, data balance, and data labeling quality of newly collected datasets may all affect the endogenous safety and security (内生安全) of machine learning models.
b) Data poisoning attack: In the maintenance and upgrading stage of machine learning algorithms, and especially in the algorithm retraining process, the risk of data poisoning may be introduced. Manual tampering with part of the training set data will directly mislead the training process and cause the model training process to be unsuccessful or lead to the injection of a backdoor in the model, making it unable to produce correct prediction results for misleading examples. This will cause model and algorithm problems.
c) Data privacy risks: The addition of new data as well as the replacement and updating of data both involve a data storage process. During this process, it is necessary to ensure the controllability of the data and avoid manual data manipulation and corruption.

B.6.3 Environment-Level Security Risks

a) Similar to the deployment and operation stage, when the framework is updated or changed or third-party libraries are updated, if corresponding adjustments are not promptly made to the models deployed on this framework, they will have configuration problems and other risks so that the models cannot be used normally.
b) During the maintenance and upgrading process, if the operating system, hardware architecture, or hardware configuration is updated, but the model is not promptly updated in time, this will lead to compatibility problems, such as insufficient precision in model processing and insufficient compute.

B.7 Decommissioning and Removal Stage Security Risks

The decommissioning and removal stage requires the destruction of the corresponding data and models. This stage mainly includes security risks at the model and data levels.

B.7.1 Algorithm-Level Security Risks

a) Risk of model and algorithm leakage: Due to an improper model destruction process, the risk of model and algorithm leakage may arise.
b) Risk of model parameter leakage: Due to the different methods used to store model parameters, different models have different destruction methods, which

may cause model parameters to remain in memory, resulting in the risk of parameter leakage.

c) Risk of incomplete model destruction: Models involve many configuration files, model parameters, and other resources, which may give rise to the risk of incomplete destruction of the model.

d) Risk of multiple devices not simultaneously destroying the model: Some models are deployed on multiple devices at the same time and are deployed through cloud-edge collaboration, so they may face the risk of incomplete destruction during destruction.

e) Risk of accidental or malicious model deletion: Due to improper permission (权限) settings, the risk of accidental or malicious model deletion may arise.

B.7.2 Data-Level Security Risks

a) Risk of incomplete data destruction: Training data, test data, and data in actual scenarios may remain in the memory and backup hard drives, resulting in incomplete data destruction.

b) Risk of private data leakage: In the process of data destruction, the risk of private data leakage may arise due to insufficient destruction means and intensity.

c) Risk of accidental or malicious data deletion: Due to improper data permission settings, some of the actually collected data and training data may be accidentally or maliciously deleted, resulting in difficulty in model reproduction.

Appendix C
(for Reference)
Adversarial Example Attacks

C.1 Adversarial Examples

Adversarial examples refer to a class of artificially constructed examples. By adding specific perturbations to original examples, a classification model is caused to make incorrect classification judgments on newly constructed examples. Many existing machine learning algorithms are highly vulnerable to adversarial example attacks. The essential reason for the generation of adversarial examples is that most current machine learning models perform mechanical data fitting and cannot understand the elements of the tasks to be performed like humans can. When the training examples cannot cover all the example spaces, the boundaries of the trained models are inconsistent with the real decision boundaries, forming a space for adversarial examples.

C.2 Objectives of Adversarial Attacks

The major objective of machine learning model adversarial attacks is to cause model output errors through minor perturbations. According to their final difficulty and impact scale, attack objectives can be roughly divided into the following four situations:

a) Incorrect prediction without target: Make the model output incorrect prediction results. That is, by adding perturbations, the output of the model is made to be different from the preset output.

b) Incorrect prediction with target: By adding perturbations, the output of the model is made to be a specific incorrect result.

c) Incorrect prediction with source and target: By adding perturbations, the model is made to realize targeted incorrect prediction for specific inputs.

C.3 Types of Adversarial Attacks

Adversarial attacks refer to characteristic attacks performed by attackers during model use by adding perturbations to the model. These attacks can be divided into the following different types based on the model information that can be obtained by the attacker:

a) White-box attacks: The network structure and network weight parameters of the machine learning model and the training data used to train the model can be obtained. That is to say, the attacker can obtain almost all the data of the model, including the loss function, the parameters obtained in final model training, and the training method. In white-box attack scenarios, the probability of model deception can still reach 100% with a small perturbation,

and most defense methods still cannot solve the machine learning model and algorithm security problems in white-box attack scenarios.

b) Gray-box attacks: The attacker can obtain the basic structure of the model, but does not know the specific model parameters; or the attack can obtain the example data used for training, that is, the distribution of the model training set can be known, but the specific structure of the model is not known.

c) Query-based black-box attacks (黑盒有查询攻击): In common applications, machine learning models and algorithms are not exposed to attackers, but attackers can usually obtain the model prediction results for any input. For example, for the facial recognition APIs developed by various companies, users can upload images to the server where the model is located, and these APIs will return prediction results. In this scenario, the attacker cannot obtain the target model, but can estimate the operation mode of the target model through access methods, and use the access results as experience to gradually enhance the attack effect, thereby achieving a black-box attack.

d) Limited-query black-box attacks (黑盒无查询攻击): One of the most difficult attack scenarios is the black-box attack method when multiple access is prohibited. For example, when machine learning models and algorithms are applied to specific applications such as face unlock and facial-recognition payment, users cannot access the target model many times, so it is necessary to crack the target model based on a limited-access (无访问) black-box attack method. When this method is required, attackers generally use attacks that exploit the transfer performance (迁移性能) of adversarial examples. Transferability (迁移性) means that the adversarial examples constructed for one model will often deceive other black-box models. Therefore, the attacker does not need to access the target model. Instead, the attacker just needs to attack a similar locally constructed model, and the generated adversarial examples can deceive the target model.

C.4 Adversarial Attack Methods

a) Fast gradient attacks (快速梯度攻击) are a type of attack method that is widely used at present. Its main idea is to find the direction in which the gradient of the deep learning model changes the most and add image perturbations according to this direction, causing the model to produce incorrect classifications. The Fast Gradient Sign Method (FGSM) perturbs images in a way that increases the loss of image classifiers. The advantage of constructing adversarial examples through fast gradient attack is the relatively high efficiency. The ultimately generated adversarial examples will cause some

slight perturbations to all pixels of the original image. As a typical attack method, many other attack methods are derived based on this method.

b) Iterative attacks: Fast gradient attacks only add one step of perturbation along the direction of gradient increase. In contrast, iterative attacks introduce multi-step small perturbations along the direction of gradient increase through an iterative method and recalculate the gradient direction after each small step. Compared with fast gradient attacks, iterative attacks can construct more precise perturbations, but at the cost of increasing the calculation volume. At the same time, they often result in overfitting.

c) Momentum iterative attacks (动量迭代攻击): Momentum-driven iterative attacks are a typical method to better balance attack performance and generalizability. A momentum mechanism is introduced in each round of iteration to reduce the shock that would occur in the original iteration. This helps the algorithm escape from local optimums and thereby converge on the global optimum or a better local optimum.

C.5 Defense Measures

To confront the above security threats to machine learning models and algorithms, machine learning model and algorithm training and testing can be strengthened so that it is more difficult to construct adversarial examples, thus making machine learning models and algorithms more robust. Specifically, the following methods can be tried:

a) Adversarial training. Adversarial training can add adversarial examples generated by various known attack methods to the training set for retraining. Adding adversarial examples during training to simulate possible data splits during testing reduces the model's error rate in identifying adversarial examples, making the final model resistant to adversarial example attacks.

b) Defensive distillation. Targeting security threats to machine learning models and algorithms, some scholars in academic circles have proposed defense techniques such as defensive distillation, adversarial training, and input reconstruction (输入重构) to combat evasion attacks. Defensive distillation connects multiple deep neural networks in a series and uses the classification results generated by the deep neural networks that come first to train the deep neural networks that come next. This reduces the sensitivity of the machine learning models to input perturbations and improves the model's stability.

c) Input reconstruction method. This method deforms input examples by adding noise, denoising, and other methods. This deformation will not affect the normal classification of the model, but allows it to resist adversarial examples

to a certain extent.

d) Strongly supervised learning (强监督学习). Traditional supervised learning algorithms generally use end-to-end learning. For example, when training a neural network, they minimize the overall (nonlinear transformation part + linear classifier part) cross-entropy in network predictions. The strongly supervised learning method imposes constraints on features learned in the nonlinear transformation part of the training process. This makes the learned features more robust to adversarial attacks and does not affect the classification accuracy of normal examples after the linear classifier part is incorporated.

e) Adversarial example detection. An adversarial example detector is constructed based on the inconsistent data distributions between adversarial examples and real examples and used to distinguish normal examples from adversarial examples. Academic circles have proposed that adversarial example detection be performed on input domains, such as the image space, feature space, and gradient space. Generally, due to the high misjudgment rate of adversarial example detectors, during actual defense, they are often used in combination with preprocessing methods such as denoising and reconstruction.