ETSI TR 104 031 V1.1.1 (2024-01)

Securing Artificial Intelligence (SAI);
Collaborative Artificial Intelligence

Reference
DTR/SAI-003

Keywords
artificial intelligence, model, trust

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Securing Artificial Intelligence (SAI).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document describes collaborative Artificial Intelligence (AI) from securing AI perspectives. Collaborative AI could take place among AI agents, between AI agents and human, and even among people who provide and use AI. As such, the security and performance of collaborative AI may range from AI/ML-specific issues to other system-specific issues (e.g. AI-to-AI communications, joint computing and communicating optimization, etc.). The present document investigates collaborative AI use cases and involved technical aspects, and analyses potential security and performance issues (e.g. AI-to-AI communications, trustworthy collaboration, etc.) among those AI-related entities. The present document also overviews existing approaches to tackle and/or mitigate these issues.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI GR SAI 009 (V1.1.1): "Securing Artificial Intelligence (SAI); Artificial Intelligence Computing Platform Security Framework".

[i.2] J. Urbanek: "Introducing Mephisto: A new platform for more open, collaborative data collection", March 2022.

[i.3] Q. Yang, Y. Liu, T. Chen and Y. Tong: "Federated Machine Learning: Concept and Applications", ACM Transactions on Intelligent System and Technology, vol. 10, no. 2, March 2019.

[i.4] K. Zhang, Z. Yang, T. Başar: "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms", April 2021.

[i.5] R. Sim, Y. Zhang, M. C. Chan and B. Low: "Collaborative Machine Learning with Incentive-Aware Model Rewards", International Conference on Machine Learning (ICML), 2020.

[i.6] N. Wang, Y. Duan and J. Wu: "Accelerate Cooperative Deep Inference via Layer-wise Processing Schedule Optimization", 2021 International Conference on Computer Communications and Networks (ICCCN), 2021.

[i.7] P. Barham, A. Chowdhery, J. Dean, et al.: "Pathways: Asynchronous Distributed Dataflow for ML", Proceedings of the 5th MLSys Conference, Santa Clara, CA, USA, 2022.

[i.8] F. Zhuang, Z. Qi, et al.: "A Comprehensive Survey on Transfer Learning", June 2020.

[i.9] ETSI GR SAI 004 (V1.1.1): "Securing Artificial Intelligence (SAI); Problem Statement".

[i.10] ETSI GR SAI 002 (V1.1.1): "Securing Artificial Intelligence (SAI); Data Supply Chain Security".

[i.11] C. Xie, M. Chen, P. Chen and B. Li: "CRFL: Certifiably Robust Federated Learning against Backdoor Attacks", in Proceedings of the 38th International Conference on Machine Learning, PMLR 139:11372-11382, 2021.

[i.12] Z. Yang, Y. Shi, Y. Zhou, Z. Wang and K. Yang: "Trustworthy Federated Learning via Blockchain," in IEEE Internet of Things Journal, 2022, doi: 10.1109/JIOT.2022.3201117.

[i.13] H. Kim, J. Park, M. Bennis and S. -L. Kim: "Blockchained On-Device Federated Learning", in IEEE Communications Letters, vol. 24, no. 6, pp. 1279-1283, June 2020, doi: 10.1109/LCOMM.2019.2921755.

[i.14] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, G. Sivastava: "A Survey on Security and Privacy of Federated Learning", Future Generation Computer Systems, Volume 115, 2021, pp. 619-640.

[i.15] B. Wang, Y. Yao, B. Viswanath, H. Zheng, and B. Y. Zhao: "With Great Training Comes Great Vulnerability: Practical Attacks against Transfer Learning", in 27th USENIX Security Symposium (USENIX Security 18), pp. 1281-1297, 2018.

[i.16] M. Xu, D. T. Hoang, J. Kang, D. Niyato, Q. Yan and D. I. Kim: "Secure and Reliable Transfer Learning Framework for 6G-Enabled Internet of Vehicles", in IEEE Wireless Communications, vol. 29, no. 4, pp. 132-139, August 2022.

[i.17] Y. Gao, and Y. Cui: "Deep Transfer Learning for Reducing Health Care Disparities Arising from Biomedical Data Inequality," Nature Communications 11, no. 1, 2020.

[i.18] S. A. Seshia, D. Sadigh and S. S. Sastry: "Toward Verified Artificial Intelligence", Communication of ACM, 65(7), pp. 46-55, July 2022.

[i.19] K. Xu, H. Ding, L. Guo and Y. Fang: "A Secure Collaborative Machine Learning Framework Based on Data Locality", IEEE Global Communications Conference (GLOBECOM), pp. 1-5, December 2015.

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

6G	Six Generation
AI	Artificial Intelligence
AIA	AI Agent
AIA4IK	AI Agent for Inferring Knowledge
AIA4LM	AI Agent for Learning a Model
AIA4LI	AI Agent for Learning and Inference
AIA4PD	AI Agent for Provisioning Data
AIH	AI Host
B-FL	Blockchain-based FL
CRFL	Certifiably Robust Federated Learning
FL	Federated Learning
FLC	FL Client
FLS	FL Server
GR	Group Report
GPS	Global Positioning System
IID	Independent and Identical Distribution

IoV	Internet of Vehicles
MARL	Multi-Agent Reinforcement Learning
ML	Machine Learning
RL	Reinforcement Learning
SAI	Securing Artificial Intelligence
TL	Transfer Learning

4 Overview

4.1 Introduction

An AI system usually contains one or multiple AI Agents (AIAs), which learn and/or exploit an AI model based on different AI schemes such as deep learning, federated learning, reinforcement learning, and/or a combination of them. AI agents usually reside in different physical or logical nodes (e.g. devices, servers, a virtual machine in the cloud), referred to as AI Hosts (AIHs) (see Figure 4.1-1). The concept of "AIHs" is consistent with the concept of "AI computing platform" described in ETSI GR SAI 009 [i.1]. Each AI agent usually hosts and runs an AI task, which could be a task for learning an AI model according to an AI algorithm (e.g. a deep learning algorithm, a federated learning algorithm, a reinforcement learning algorithm) or a task for using an AI model to infer knowledge. Deep learning and reinforcement learning usually uses one AI agent, while federated learning utilizes multiple AI agents working collaboratively to learn an AI model. An AI algorithm could be supervised by relying on tagged training data or unsupervised without the use of any tagged data.

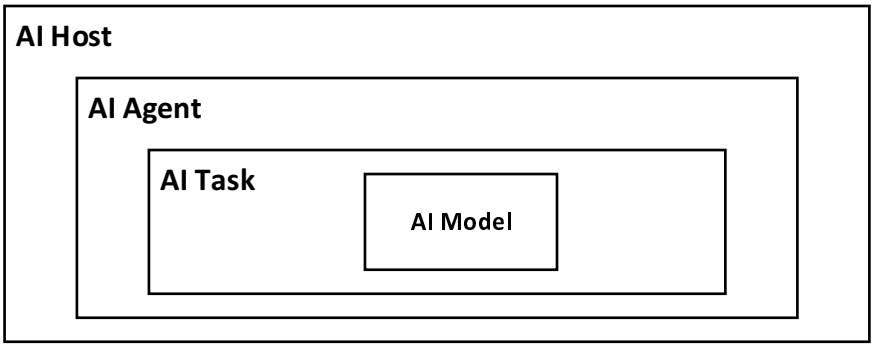

Figure 4.1-1: AI Host, AI Agent, AI Task, and AI Model

4.2 AI pipeline

A general AI pipeline for supervised learning is illustrated in Figure 4.2-1, which usually consists of multiple stages:

1) task configuration stage that includes the deployment of AI agents/tasks by an AI application or user;
2) data preparation stage that includes data collection and optional feature engineering/extraction;
3) training stage for learning an AI model;
4) validation stage for testing and validating the learned AI model;
5) model deployment stage for deploying and transferring the validated AI model; and
6) inference stage for inferring and predicting future knowledge using new data as inputs (referred to as input data for inference).

The outcome/results from the inference stage could be leveraged to action or trigger going back to training stage to re-train the AI model.

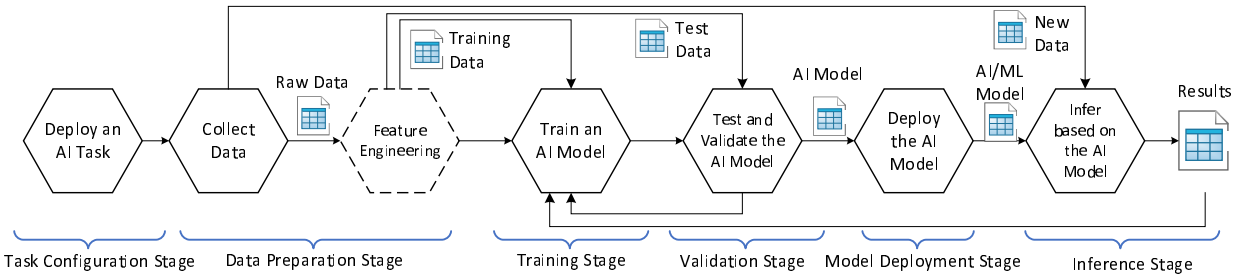

Figure 4.2-1: General AI Pipeline for Supervised Learning

Dependent on AI deployment choices, an AI agent could be:

1) An AI Agent for Learning a Model (AIA4LM) that is only responsible for learning an AI model;

2) An AI Agent for Inferring Knowledge (AIA4IK) that uses a learned AI model for inference and predication; and

3) An AI Agent for Learning and Inference (AIA4LI). AI model transfer generally occurs between an AIA4LM and an AIA4IK or between multiple AIA4LIs.

4.3 Collaborative AI

Collaborative AI can be classified to the following categories:

- Category 1: Agent-to-Agent Collaboration. In this category, more than one AI agent collaborate with each other during the partial or the whole AI pipeline to perform an AI task. Examples of this collaborative AI category include collaborative data collection [i.2], federated learning [i.3], multi-agent reinforcement learning [i.4], collaborative machine learning [i.5], collaborative inference [i.6], Pathway as next-generation AI [i.7].
- Category 2: Agent-to-Human Collaboration. In this category, AI agents and human collaboratively work together to solve a shared task.
- Category 3: Collaborative AI Marketplace. In this category, human collaboratively exchanges and shares data, training capability, AI models, and/or inferred knowledge via an open AI marketplace.

5 Use cases

5.1 Introduction

Clause 5 describes three categories of collaborative AI use cases, which are collaborative distributed AI/ML, Human-AI collaboration, and collaborative AI/ML marketplace.

5.2 Collaborative distributed AI/ML

5.2.1 Federated Learning (FL)

Federated Learning (FL) [i.3] is a framework for distributed machine learning, where a FL Server (FLS) and FL Clients (FLCs) collaboratively learn an AI model. In FL, training data is maintained locally at multiple distributed FLCs (e.g. mobile devices). Each FLC performs local training (e.g. deep learning), generates local model updates, and sends local model updates to the FLS. The FLS as a central entity aggregates local model updates received from FLCs and generates global model updates, which will be sent to all participating FLCs for the next training round. In fact, the FLS hosts an AI agent for learning, while each FLC has an AI agent that could be for both learning and inferring.

Figure 5.2.1-1 illustrates the general federated learning process, where the FLS and FLCs jointly take the following steps to perform an FL task:

- step 1: The FLS selects a set of FLCs to participate in a FL task;
- step 2: The FLS configures the FL task to each selected FLC;
- step 3: The FLS sends an initial global model to each selected FLC;
- step 4: Each FLC independently trains the global model based on the received initial global model and its local data;
- step 5: After each training round, each FLC generates a local model update and sends it to the FLS;
- step 6: The FLS receives local model updates from all FLCs, aggregates them, and generates new global model update. The FLS may need to wait for receiving local model updates from all FLCs before performing the aggregation (i.e. synchronous FL) or it can start the aggregation after receiving the local model updates from some of FLCs (i.e. asynchronous FL). Note that the FLS may (re)select some new FLCs for next training round;
- step 7: Similar to Step 3, the FLS sends the global model updates to all FLCs; and
- step 8: Similar to Step 4, each FLC starts next local training.

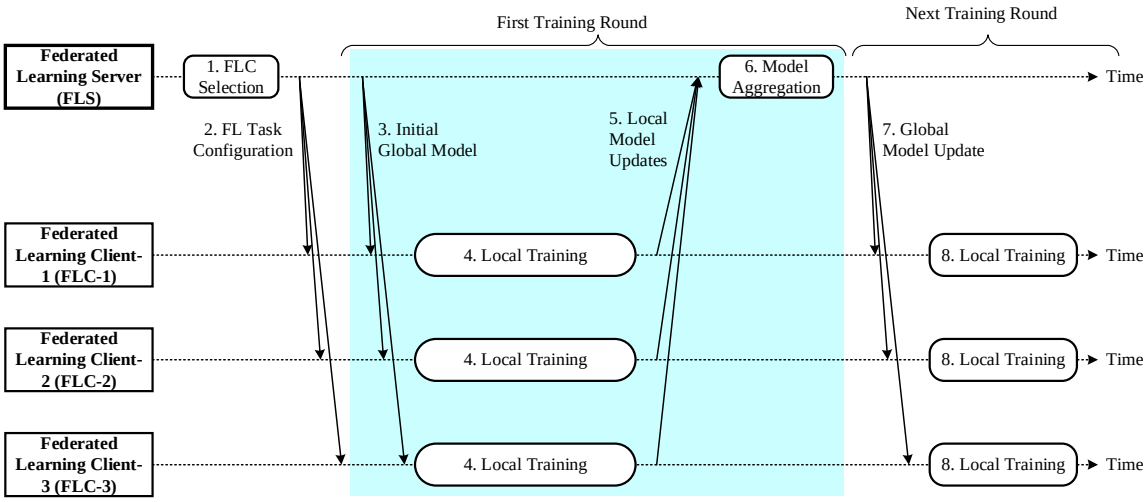

Figure 5.2.1-1: Federated learning

Several advantages of federated learning include:

1) improved data privacy-preservation since training data stays at FLCs;
2) reduced communication overhead since it is not required to collect/transmit training data to a central entity; and
3) improved learning speed since model training now leverages distributed computation resources at FLCs.

However, FL needs to transmit model updates among AI agents (i.e. between the FLS and FLCs), which introduces security issues and additional communication overhead compared to centralized machine learning. Also, FL requires data at all FLCs follow an Independent and Identical Distribution (IID) (i.e. IID-data) in order to achieve a good learning performance. In addition, FL inherits some potential security issues and threats such as data poisoning and model poisoning attacks.

There are many real scenarios where FL is used to learn a global AI model without collecting training data from end devices and/or protecting data privacy. For example, an end device such as a smart phone nowadays has more sensing capabilities (e.g. camera, sensors), which generate different types of data streams. A global AI model can be learned from these data streams from different end devices. Instead of collecting such data streams from end devices to cloud, each end device performs local training independently and reports its local model updates to an FLS, which aggregates model updates received from multiple end devices and produce the global AI model.

5.2.2 Transfer Learning (TL)

Transfer Learning (TL) has attracted much attention in recent years [i.8] due to the fact that many AI tasks are more or less relevant. In general, TL uses the source AI model trained from the source domain as a starting point to train the target AI model in the target domain. For example, the model parameters or the model structure of the pre-trained source AI model is transferred from the source AI agent to the target AI agent and is used by the target AI agent to train the new target AI model in the target domain (see Figure 5.2.2-1).

There are several TL methods such as Pre-train and Fine-tune, Domain Adaptation, Domain Generalization, and Meta-learning. Taking the Pre-train and Fine-tune method as an example, the first step is to find a pre-trained AI model that can be used for the new problem. It is important to choose a pre-trained AI model carefully.

EXAMPLE: An AI model for riding a bicycle cannot be used for training a self-driving car model and maintain trust.

After a pre-trained AI model is determined, there are usually several approaches to transfer and leverage the pre-trained AI model, such as:

- To remove the output layer of the pre-trained AI model and use it as a feature extractor for the new training data from the target domain.
- Another approach is to transfer the structure of the pre-trained AI model, but re-train all the weights with the new training data from the target domain.
- A different approach is to re-train specific layers but reuse other layers of the pre-trained AI model. For example, for a neural network model, some lower layer of the network (which are used to identify the underlying features of various objects such as boundaries and shapes) can be reused as they are, while only some higher layers (which are used to identify advanced features such as the specific appearance of the face) will be retrained.

TL requires communications between AI agents (i.e. the AI model knowledge transfer from the source AI agent to the target AI agent), which introduces security issues. In addition, since TL relies on the pre-trained AI model, it inherits potential threats to the pre-trained AI model (e.g. data poisoning and backdoor attacks to the pre-trained AI model).

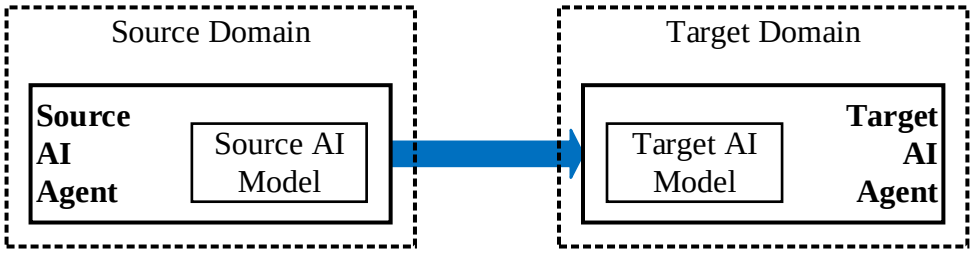

Figure 5.2.2-1: Transfer learning

5.2.3 Multi-Agent Reinforcement Learning (MARL)

Different from traditional supervised or unsupervised machine learning, Reinforcement Learning (RL) is a continuous machine learning paradigm, often used to solve sequential decision-making problems. An RL agent keeps interacting with the environment to gain real experience (i.e. samples); in the meantime, it keeps learning from the gained real experience an optimal policy, which is usually used to control or influence the environment and in turn new real experience will be generated.

The optimal policy that an RL agent learns or determines is actually a policy function $\pi(s, a)$, which maps from state space S to action space A. The RL agent can also indirectly determine the optimal policy for next step based on estimating the following two value functions:

1) $V(s)$ is the state value function that calculates the average one-step reward the RL agent can collect by taking all possible actions at current state *s*; and

2) $Q(s, a)$ is the state-action value function which calculates the average one-step reward the RL agent can collect by taking an action *a* at current state *s*.

RL can be performed in a distributed and parallel way, referred to as Multi-Agent Reinforcement Learning (MARL) [i.4]. Under MARL, multiple RL agents usually interact with the same environment and learn the optimal policy collaboratively or independently/competitively. Figure 5.2.3-1 illustrates general MARL architecture using three RL agents as an example, where each RL agent performs the following operations:

- Each RL agent observes the environment to obtain real experience, based on which it learns the optimal policy. Then, the RL agent determines a new action according to the learned optimal policy and takes the new action.
- The new action taken by each RL agent will impact the same environment and accordingly impact new observations of other RL agents. In other words, even though each RL agent may independently learn the optimal policy and independently take a new action, their observations on the same environment will be impacted by the action taken by any RL agent.
- Multiple RL agents may share the same objective. As such, they collaboratively learn the optimal policy together, which can maximize (or minimize) their common objective function. To this purpose, they may exchange and share some information such as their observations about the environment, their learned policy, their actions being taken, etc.
- Each RL agent may have a different objective, for example, to maximize the total or average reward for itself. In this case, each RL agent behaves like an independent RL agent and may not exchange any information with other RL agents, unless some of them establish an agreement to share some limited information (e.g. local observations at a RL agent).

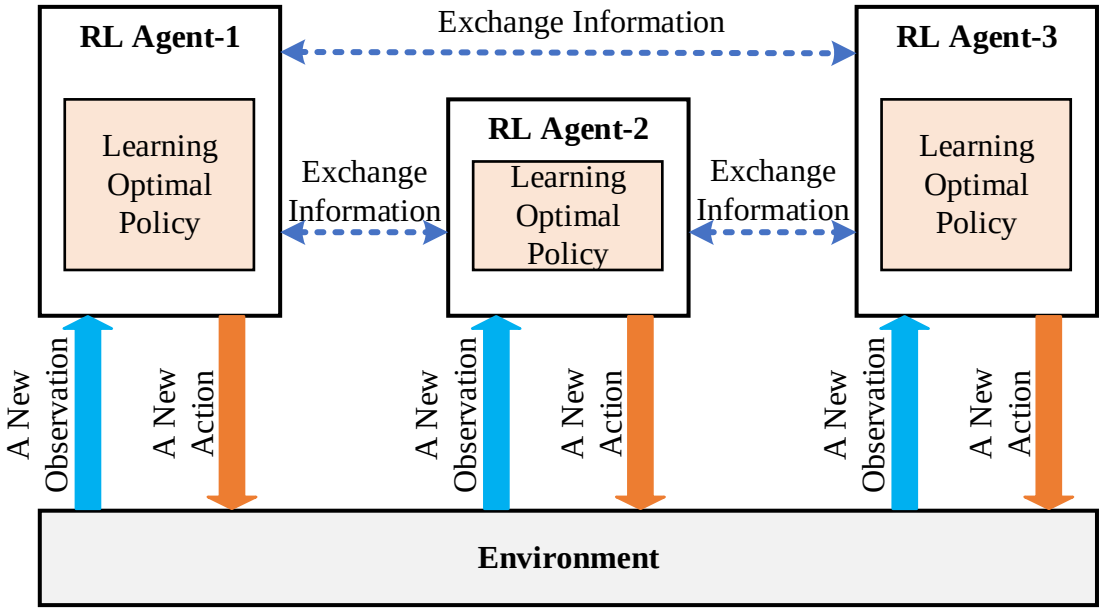

Figure 5.2.3-1: General MARL architecture

5.2.4 Pathways

Pathway [i.7] is proposed as a next-generation AI architecture, where multimodal AI models can be leveraged by each other to fulfil a variety of AI tasks through an asynchronous distributed dataflow design. In Pathway, there are multiple AI agents, which can be sequentially combined and reused in different ways for solving more complicated AI tasks. For example, an AI task A may use concatenated three AI agents: AI-Agent-A, AI-Agent-B, and AI-Agent-C, which forms an AI path A; another AI task B may use other four AI agents sequentially: AI-Agent-A, AI-Agent-C, AI-Agent-D, and AI-Agent-E, which forms an AI path B. On each AI path, an AI agent in the middle of the path will take the output from previous AI agent as its input, runs its local operations, generates its output, and forwards its output to immediately next AI agent on the AI path.

Pathways could be useful for many scenarios such as autonomous driving. For example, an autonomous car may need to handle multiple driving tasks on the road, such as traffic light recognition, pedestrian detection, speed control, forward collision avoidance, left turn assistance, vehicle platooning, etc. Traditionally, those tasks may need different AI models in the sense that each AI model is trained individually. However, such an approach is not like the way how human brain makes driving actions and maneuvers the car on the road. In Pathways, a general AI model can be trained for the autonomous driving, which can handle multiple driving tasks at the same time. In this way, the AI model associated with each task does not have to be trained from scratch. Instead, the general AI model may work on different tasks by using multi-modality inputs collected from road (images, sensory, radar, etc.) and those tasks may also cooperate with each other and provide useful knowledge. For example, a speed control task is to decide what speed a vehicle should maintain when passing an interaction. In this case, the speed control task may obtain useful inputs from traffic light detection task (what is the traffic light currently), pedestrian detection task (whether there is a pedestrian crossing the intersection), and traffic flow detection (what is the current moving speed of other surrounding vehicles), in order to decide a best moving speed for the current vehicle.

5.3 Human-AI collaboration

Another type of collaborative AI is related to Human-AI collaboration. Although AI/ML technology has made great progress, machine intelligence still cannot completely replace human involvement for the time being. For example, there are many application scenarios where people and machines need to work together, such as:

- In big data analysis application scenarios, AI can help people in completing various complex data processing tasks (e.g. processing bills, performing massive information mining, conducting predictive analysis, etc.). However, some high-level decisions and strategy formulation still mainly rely on human participation.

- In a smart factory, workers may need to work with assembly robots. Robots need to respond timely and correctly to various human behaviours, which are sometime even unexpected actions. In particular, it is also important to make sure any robot action does not cause potential harm to the workers.

- In autonomous vehicle driving, the AI driving module can reduce the driving burden of human drivers and enable the vehicle to drive safely under normal road conditions. But the collaboration between the AI driving module and the human driver is needed. On one hand, in special or unexpected abnormal road conditions that the AI driving module can sense but may be incapable to handle, the AI driving module can send explicit indications or hints to the human driver so that the human driver can take over the driving task at the first time. On the other hand, the human driving can configure certain driving instructions and/or policies (e.g. speed limits) to the AI driving module to make the autonomous driving more regulated and safer. In another example, some autopilot systems sometimes require user inputs to steer them towards a desirable decision (e.g. pressing the gas pedal at a particularly complex intersection where the autopilot system is unsure whether to turn right or proceed forward, given that both paths might be valid according to the Global Positioning System (GPS) and the given driving situation).

5.4 Collaborative AI/ML marketplace

Another type of collaborative AI is about the collaboration and cooperation of multiple people for their AI tasks, for instance, through an AI/ML marketplace. This is because, in many different AI tasks, people often encounter similar challenges, such as how to choose the most suitable training data and algorithms for a specific model training problem. If people can learn from the experience of others, it can save a lot of time for their own AI training tasks. One way of doing this is to use a sharing platform that allows people to share AI models, AI training data, AI experience, etc. For example, users can upload their own datasets, AI training workflow/process, and AI experiment results. At the same time, users can also discover the desired datasets shared by others, search for other people's previous experimental results, and/or learn from other's previous experience to set up their new AI model training experiments.

6 Security threats

6.1 Introduction

Clause 6 describes potential security threats in collaborative AI in the following scenarios:

- AI-to-AI communications
- Attack Propagation
- Collaborative data supply chain
- Trustworthy collaboration
- Audition and non-repudiation

6.2 AI-to-AI communications

Multiple types of AI agents may exist in an AI system such as AI Agent for Provisioning Data (AIA4PD), AI Agent for Learning a Model (AIA4LM), and AI Agents for Inferring Knowledge (AIA4IK). Under collaborative AI, the same or different types of multiple AI agents communicate with each other to exchange some information, which could be training data, model updates, learned models, input data for knowledge inference, and/or inferred knowledge. Such AI-to-AI communications may include the following scenarios:

- **AIA4PD-to-AIA4LM Communications:** An AIA4PD provisions and sends training data to one or multiple AIA4LMs. Alternatively, an AIA4LM requests and retrieves training data from one or multiple AIA4PDs.
- **AIA4LM-to-AIA4LM Communications:** Multiple AIA4LM exchange AI models (e.g. in federated learning) and/or even training data.
- **AIA4LM-to-AIA4IK Communications:** An AIA4LM sends an AI model to one or multiple AIA4IK. Alternatively, an AIA4IK requests and retrieves an AI model from one or multiple AIA4LM.
- **AIA4IK-to-AIA4IK Communications:** Multiple AIA4IK exchange a portion of the AI model (and even input data for inference) to enable collaborative knowledge inference.

In AIA4PD-to-AIA4LM Communications, there are several security risks and threats:

- **Communication-Level Attack:** A malicious node may congest or attack the communication link between AIA4PD(s) and AIA4LM(s). This type of attacks threatens secure and reliable communications between AIA4PD(s) and AIA4LM(s).
- **Data-Level Attack:** A malicious AIA4PD may provide inaccurate or tampered data to AIA4LM, which is a kind of data poisoning attacks.

- **Node-Level Attack:** An AIA4LM could also become a malicious node in the sense that it may fully control an AIA4PD and prevent the AIA4PD from serving other AIA4LMs. As an example, an AIA4PD could be a shared outdoor camera, which may change its viewport/orientation based on the needs of AIA4LMs. A malicious AIA4LM (e.g. for training an AI model for image/video recognition) may try to infect the AIA4PD with a malware so that this AIA4PD cannot serve other AIA4LMs but this malicious AIA4LM.

In AIA4LM-to-AIA4LM Communications, there are several security risks and threats:

- **Communication-Level Attack:** A malicious node may congest or attack the communication link between AIA4LMs. In FL, a malicious FL client (as an AIA4LM) may initiate a network attack on other FL clients (as AIA4LMs) so that other FL clients may not be able to upload their local model updates to the FL server for model aggregation. This type of attacks threatens secure and reliable communications between AIA4LMs.

- **Model-Level Attack:** Under FL, FL clients as AIA4LM conduct local training using local data and then submit their local model updates to the FL server (also an AIA4LM) for model aggregation. If an FL client is malicious, it can upload bad local model updates to the FL server in order to contaminate the global model and downgrade its accuracy.

- **Node-Level Attack:** An attacker may initiate an attack on the FL server. For example, the attack may affect the model aggregation operation at the FL server so that the global model training will never be converged. For example, the attacker may make the FL server send old or wrong global model updates to FL clients so that those FL clients will repetitively conduct useless local training.

- **Model-Level Attack:** In transfer learning, an AIA4LM-1 may need to obtain a trained model from another AIA4LM-2 as its training start point. In such a case, if AIA4LM-2 is a malicious node, it may send a bad AI model to AIA4LM-1 such that the model to be trained by AIA4LM-1 will also inherit malicious information or characteristics.

In AIA4LM-to-AIA4IK Communications, there are several security risks and threats:

- **Communication-Level Attack:** A malicious node may congest or attack the communication link between AIA4LM(s) and AIA4IK(s). This type of attacks threatens secure and reliable communications between AIA4LM(s) and AIA4IK(s).

- **Model-Level Attack:** If an AIA4LM is malicious, it may deliver and deploy a bad AI model to an AIA4IK. In the example of autonomous driving, an AIA4LM may deliver a road sign detection and recognition AI model onto an autonomous car (as an AIA4IK). In such a case, the autonomous car may mistakenly recognize road signs, which may lead to serious traffic accidents.

In AIA4IK-to-AIA4IK Communications, there are several security risks and threats:

- **Communication-Level Attack:** A malicious node may congest or attack the communication link between AIA4IKs. This type of attacks threatens secure and reliable communications between AIA4IK(s).

- **Model-Level Attack:** If an AIA4IK is a malicious node, it may plant certain malicious information into the model delivered by the AIA4LM, and then further shared this infected or malicious AI model with other AIA4IKs.

- **Data-Level Attack:** When AIA4IKs need to exchange input data for collaborative knowledge inference, a malicious AIA4IK (or other attackers) may provide inaccurate or tampered input data to other AIA4IKs for data poisoning attacks.

6.3 Attack propagation

ETSI GR SAI 004 [i.9] describes several attack types to AI system including poisoning, input attack and evasion, backdoor attacks, and reverse engineering. Under collaborative AI, those attacks may be propagated from one AI agent to other AI agents and continues especially when multiple AI agents form a chain structure. Using Figure 6.3-1 as an example, an original input attack to AI agent A will lead to a wrong output, which if leveraged by AI agent B as its input will lead to another wrong output (i.e. a propagated attack). When AI agent C takes the wrong output from AI agent B as its input, AI agent C will generate a wrong output too (i.e. a propagated attack). Similarly, when those AI agents collaboratively train an AI model, a poison attack to AI agent A could impact other two AI agents as well.

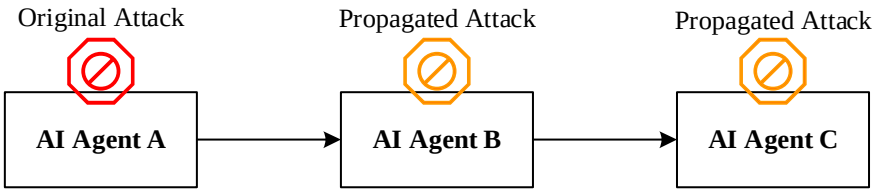

Figure 6.3-1: Propagated attacks in collaborative AI

6.4 Collaborative data supply chain

ETSI GR SAI 002 [i.10] describes data supply chain for AI systems, which generally consists of several data phases: data acquisition, data curation, training and testing, deployment, and data exchange. Data integrity could be attacked and lost during any of those data phases. In collaborative AI, those data phases could be distributed in different AI agents. As a result, data integrity could be lost at each AI agent or in the transit from one AI agent to another AI agent.

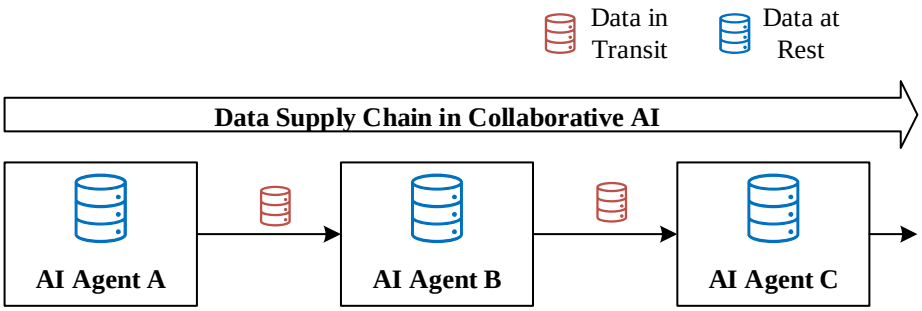

Figure 6.4-1: Data supply chain in collaborative AI

6.5 Trustworthy collaboration

In collaborative AI, many AI agents interact with each other to achieve collaborative training or collaborative inference. For this purpose, they need to collaboratively exchange training data, the model, and/or inferred knowledge. Those AI agents may belong to different organizations and do not have any pre-established trust relationships. To secure collaborative AI, it is critical to guarantee the trust and trustworthy collaboration among those AI agents. Note that even if some AI agents are from the same organization, they still need to build a trust relationship under zero-trust architecture.

Dependent on how those AI agents collaboratively exchange AI-related information, there could be two trust models for them: Chain of Trust and Decentralized Trust:

- **Chain of Trust:** As illustrated in Figure 6.5-1, multiple AI agents form a chain or a tree structure rooted at AI Agent A, which is a root of trust. For example, the output of AI agent A is fed to AI agent B as its input; likewise, the output of AI agent B is sent to AI agent D as its input. In this case, AI agent A is trusted by AI agent B and C; then, AI agent D trusts AI agent B and AI agent E trusts AI agent C. If a particular AI agent involved a collaborative AI task is attacked, all other AI agents that rely on this particular AI agent will be impacted. As a result, this collaborative AI task becomes untrustworthy and unsecure.

- **Decentralized Trust:** Multiple AI agents may interact with each other for a decentralized and collaborative AI task such as fully distributed federated learning and multi-agent reinforcement learning. In this case, decentralized trust among those AI agents is required as shown in Figure 6.5-2. There is no such a root of trust in decentralized trust. Compared to chain of trust, decentralized trust could have better protection or still workable even if one AI agent is attacked.

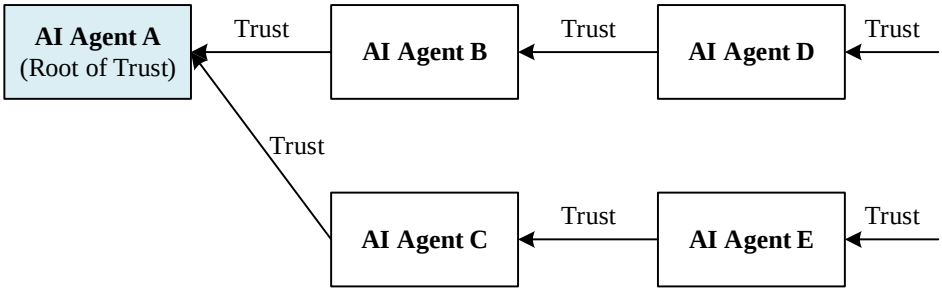

Figure 6.5-1: Root of trust and chain of trust in collaborative AI

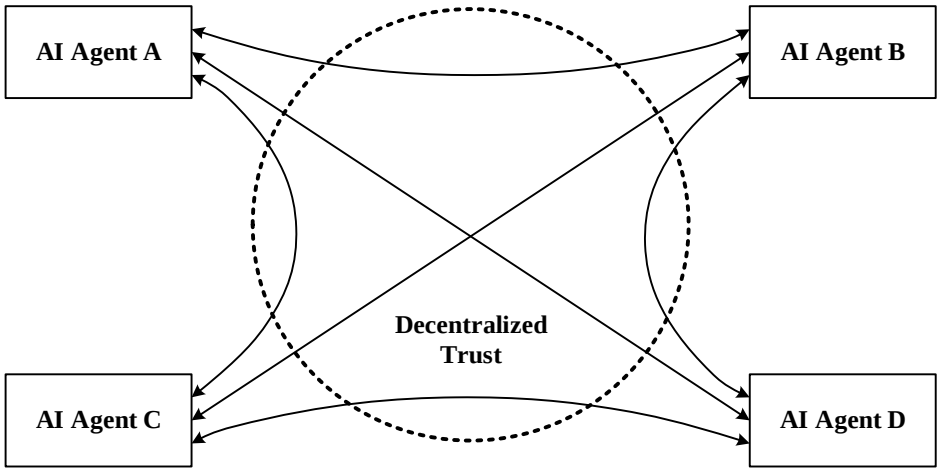

Figure 6.5-2: Decentralized trust in collaborative AI

6.6 Audition and non-repudiation

In a collaborative AI system, multiple AI agents are involved in different types of AI tasks. As such, auditing and non-repudiation mechanisms need to be in place to enable effective and safe collaboration among multiple AI agents.

For example, the information exchanged in collaborative AI scenarios may include original data generated by an AI agent (such as an AIA4PD). The data generated by AIA4PD may be provided to AIA4LMs for training purposes. A malicious AIA4PD may provide undesired data to an AIA4LM. Such undesired data may either be inaccurate or may even be poisonous. Thus, it is important to enable audition of data provisioning (e.g. to record the evident of data provisioning actions and history done by the AIA4PD). In other words, a non-repudiation mechanism, when applied to maintain certain evidence or proof, can make AIA4PD incapable of denying its data provisioning actions. Similarly, multiple AIA4LMs may also work together for the same AI training task. For example, under FL, FL clients (as AIA4LMs) may need to submit their local model updates to the FL server (as another AIA4LM) for model aggregation. Therefore, a non-repudiation mechanism is also needed since an FL client could be malicious and deny its malicious local model submissions.

Overall, audition and non-repudiation mechanisms are needed for securing collaborative AI. There could be different ways to support audition and non-repudiation in collaborative AI scenarios. Traditional approaches may rely on a centralized party (e.g. certification authority and digital signatures). Alternative solutions include the use of immutable distributed ledgers to record all the essential actions/information and interactions among collaborative AI agents, so that audition can be performed at any time to guarantee non-repudiation.

7 Existing solutions

7.1 Introduction

This clause briefly describes and summarizes some existing example solutions for securing collaborative AI.

7.2 Secure Federated Learning (FL)

In Federated Learning (FL), FL Clients (FLCs) keep generating local model updates and sending them to the FL Server (FLS), during which a malicious FLC can use backdoor attacks and model replacement to introduce backdoors into the trained global model. It becomes important to detect and mitigate backdoor attacks to the global model in a FL system to achieve secure FL. For example, robust model aggregation schemes at the FLS can be designed to filter out model updates from malicious FLCs. To achieve guaranteed robust FL, a Certifiably Robust Federated Learning (CRFL) framework was proposed in [i.11], which introduces no changes to FLCs but some techniques to the FLS. Specially, after regular model aggregation to generate a new global model, the FLS clips the new global model and also introduces a noise to it. Then, the FLS broads the clipped and noised global model to all FLCs for next training round. Then, during testing phase of CRFL, the FLS smooths global model parameters (again by adding noises) to generate multiple noisy global models, which each test sample will be applied to. The outputs from applying each sample to all noisy global models were used to calculate the certified radius, which is an upper bound on the distance that a poisoned model can drift from a benign model, and limits the impact that an attacker can have on the model.

Another security issue in FL is trustworthiness since FLCs could be from different administrative domains. The proposed Blockchain-based FL (B-FL) in [i.12] focused on trustworthy FL by leveraging blockchain technology. The major motivation of B-FL is that, although FL has the advantage in terms of privacy protection, there are still many other security challenges for FL such as potential malicious FL clients and/or malicious FL server. In order to enable trustworthy and effective collaboration among all the FL participants, blockchain is used to log all the essential operations and model updates for the FL training. In B-FL architecture, multiple edge devices can conduct their local training and submits the local model updates to a primary edge server. Accordingly, the primary edge server may include submitted local model updates and the aggregated global model update in a new block, which are sent to a batch of other edge servers as validators, which rely on various consensus protocols (e.g. Proof of Work or Practical Byzantine Fault Tolerance) to achieve consensus. Once consensus is achieved among all the validators for appending this new block, the aggregated global model updates included in the block will be dispatched to the edge clients for the next round of local training. [i.13] proposed a fully decentralized FL, referred to as BlockFL. The centralized FLS may have potential security issues, such as single point of failure, and is vulnerable to various security attacks. In the meantime, FLCs cannot be well incentivized for participating in the FL training, which leads to low-performance collaborative AI. As such, BlockFL was proposed as a new decentralized FL architecture. BlockFL has no FLS and the FL model aggregation can be conducted within the blockchain network.

More solutions for mitigating FL security and privacy issues can be found from [i.14].

7.3 Secure Transfer Learning (TL)

It is known that many small companies have limited computing capacity and a small set of application data available for training; therefore, producing a sophisticated/accurate AI model is often challenging. In contrast, some big companies (e.g. public cloud providers) have powerful computing infrastructures and massive data sets (e.g. millions of images available on the Internet); they can train and publish their complex and general AI models. Then, transfer learning can be leveraged for small companies to efficiently train their own AI models. For example, a highly-tuned AI model as a Teacher model (developed/published by big companies) may be further customized by small companies using additional training (e.g. based on domain-specific data sets) to generate Student models. This work in [i.15] identified a new attack for transfer learning. Since a Teacher model is often publicly-accessible and its internal presentation is often known to an attacker. In transfer learning, the first K layers of the Teacher model was usually frozen and copied into the Student model. As such, the attacker may add a small perturbation on a source image (e.g. a cat) so that it can mimic the internal representation of the target image (e.g. a dog) at layer K. The key insight is that in feedforward networks, each layer can only observe inputs from the previous layer; therefore, as long as the attacker may make the representation of the source image at layer K perfectly matching with that of the target image at layer K, the source image will be misclassified into the target image, regardless of the weights of layers after layer K. The authors claimed that this type of attacks may be very sensitive to small changes in the adversarial samples, based on which, they proposed a number of effective defence solutions. One defence solution is to introduce additional random perturbation to any input image before starting classification. Another proposed solution is to modify the weights of different layers of the Student model so that the attacker is unable to leverage the similarity between the matching layer in the Teacher and Student models.

Transfer learning could be useful but more vulnerable as well for Internet of Vehicles (IoVs). Each IoV may have limited computing capacity and limited knowledge/data about the physical driving environment. As such, knowledge sharing via transfer learning is desired compared to traditional AI/ML. For example, an IoV A with more miles basically accumulates more road-trip data covering many different risky driving and traffic accident scenarios. Therefore, the IoV A could train a good driving assistance model, which can be shared with another IoV B so that the IoV B can leverage the model shared by the IoV A as a starting point to train its own driving assistance AI model. Such an AI model for safety-related AI model has to be as accurate and reliable as possible; however, sometimes an attacker IoV may even share a malicious model with other IoVs and cause serious consequences. The work in [i.16] focused on how to design secure transfer learning for IoVs in the 6G era. The authors proposed a secure and reliable transfer learning framework for 6G-enabled IoVs. In particular, the framework includes a blockchain system for building trust among different IoVs. In the meantime, the blockchain can also be used for supporting reputation management so that any malicious behaviour during transfer learning process may be recorded and leveraged to detect attackers.

Transfer learning also has important applicability in the domain of health care. For example, a healthcare disparity issue due to biomedical data inequality was identified in [i.17]. In this research, the authors observed that data inequality among ethnic groups significantly affected the performance of AI in the healthcare system. As an example, cancer-related data in some clinical projects was mainly collected from Caucasians (91,1 %) while only 5,6 % of data was collected from Asians. As a result, non-Caucasians (around 84 % of the world population) in fact suffer from data inequality, since an ML model is often trained on the mixed data sets collected for all ethnic groups. The consequence is that the ML model may have a bad performance on the Asian group due to inadequate data. This work in [i.17] proposed to use the transfer learning to solve this problem. For example, the model trained on the data from the non-Caucasians group can be transferred/used for training a new ML model for the Asian group using their own data. In this way, the generated new model may have improved performance for the Asian group even if this group only has insufficient data. From secure AI perspective, secure model transfer should be guaranteed.

7.4 Verification of collaborative AI

Another security aspect of collaborative AI is related to verified collaborative learning system. For example, the authors of [i.18] focused on verified artificial intelligence. They advocated that the needs, as well as the detailed specification of a desired AI system, are formally described using mathematical descriptions. This is also applicable to the collaborative learning case. In the meantime, the security characteristics of the built collaborative AI system can themselves be verified via strict mathematical proof. This is very important, especially for some safety/life-critical scenarios. Overall, the formal specification of the collaborative AI system is required in order to conduct quantitative and mathematical verification.

7.5 Data privacy in collaborative AI

Data privacy is another critical issue in collaborative learning scenarios. The authors of [i.19] proposed a secure collaborative ML framework that could deal with two different scenarios:

1) horizontal data partition case where data sample sets are partitioned among different AI agents based on rows; and

2) vertical data partition case where data sample sets are partitioned among different AI agents based on columns.

In particular, this work presented a thorough security and mathematical analysis to demonstrate how the proposed solution may exchange just-in-enough information during AI-to-AI communications but other than that, nothing else is leaked.

8 Conclusions and next steps

The present document discusses collaborative AI. It first described AI pipeline and collaborative AI in general. Then a few collaborative AI use cases were presented. Security threats in collaborative AI were analysed and discussed. Some existing example solutions for securing collaborative AI were briefly summarized.

The following concepts and topics for or related to the interactions among AI agents for attacking security threats in collaborative AI could be considered for standardization:

1) Specifications for AI-to-AI communications;

2) Specifications for collaborative data supply chain; and

3) Specifications for enabling trustworthy AI collaboration with audition and non-repudiation.

History

Document history		
V1.1.1	February 2024	Publication