Maschinelles Lernen auf geheimen und/oder personenbezogenen Daten

KARL
Künstliche Intelligenz
für Arbeit und Lernen

Kompetenzzentren
Arbeitsforschung

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

Maschinelles Lernen auf geheimen und/oder personenbezogenen Daten

KARL-Gestaltungsfeld: Datenqualität, Datenschutz und Datensicherheit

Der Einsatz datengetriebener KI (Methoden des maschinellen Lernens) muss häufig mit dem Schutzbedarf der benötigten Daten in Einklang gebracht werden, beispielsweise falls die Daten Geschäftsgeheimnisse darstellen oder falls sie personenbezogen sind und daher dem Datenschutz unterfallen. In diesem Überblick werden KI-spezifische Fragestellungen der Informationssicherheit für das Training und den Betrieb von Modellen und jeweils mögliche Lösungsansätze beleuchtet.

1. Einleitung

In vielen Anwendungsdomänen, in denen der Einsatz maschineller Lernverfahren vielversprechend ist und vorangetrieben wird, sind die auszuwertenden Daten schützenswert. So hat man es etwa im Gesundheits-, Finanz- und Personalwesen mit personenbezogenen Daten zu tun, teilweise sogar mit besonderen personenbezogenen Daten mit erhöhtem Schutzbedarf. Auch in den Domänen Mobilität und Produktion können personenbezogene Daten auszuwerten sein. Meist sind hier aber auch Daten schützenswert, die nicht personenbezogen sind, da sie Betriebsinterna darstellen, in denen ggf. Betriebsgeheimnisse angelegt sind bzw. auf denen Geschäftsmodelle der betreffenden Organisationen beruhen. In der Regel ist das primäre Schutzziel, das bei der Auswertung schützenswerter Daten erreicht werden soll, deren *Vertraulichkeit*, seien es personenbezogene oder anderweitig geheime Daten.

Mit maschinellen Lernverfahren wird anhand bereits vorhandener Daten, den sogenannten Trainingsdaten, ein Modell gelernt und auf einem nicht benutzten Teil der Trainingsdaten validiert. Hiermit soll gewährleistet werden, dass das Modell im Betrieb Aufgaben erfüllen kann wie etwa neue Daten zu klassifizieren (In welches Preissegment fällt ein Haus mit den gegebenen Daten?, Ist die Person ist ein Risikopatient für Herz-Kreislauferkrankungen?, Ist die Mitarbeiterin auf dem Absprung oder nicht?) oder anhand neuer Daten eine Prädiktion abgeben (Was ist ein Haus mit den gegebenen Daten wert?, Wie hoch ist das Risiko der Patientin, an Diabetes zu erkranken?, Welches Gehalt sollte dem Mitarbeiter bezahlt werden?). Hinsichtlich des Schutzziels der Vertraulichkeit ist somit zunächst zwischen den Trainingsdaten und den Daten, die das Modell im Betrieb verarbeitet zu unterscheiden. Hierzu kommen jeweils unterschiedliche Maßnahmen in Betracht, die im Folgenden näher ausgeführt werden. In Szenarien, in denen während des Betriebs eines Modells Daten zum Nachtrainieren erhoben bzw. gespeichert werden, ist der Übergang zu den Trainingsdaten fließend.

Weniger eindeutig zu beantworten ist die Frage nach dem Schutzbedarf der Modelle. Einige Forschungsergebnisse zu erfolgreichen Privacy-Angriffen auf Modelle wurden bereits publiziert, bei denen sensible Informationen aus Modellen extrahiert werden konnten. Insofern wird auch die Frage diskutiert, inwiefern Modelle als personenbezogene Daten zu behandeln sind. In diesem Dokument wird ein Überblick über die Techniken von Privacy-Angriffen auf Modelle gegeben. Während es kaum allgemein zu beantworten ist, auf welche Arten von Daten und Modellen die publizierten Privacy-Angriffe im Einzelnen übertragbar sind, können Gegenmaßnahmen leichter angegeben werden. Der vielversprechendste Ansatz, *Differentially Private Machine Learning*, ist allerdings nicht ohne Einbußen bei der Performanz des resultierenden Modells einsetzbar.

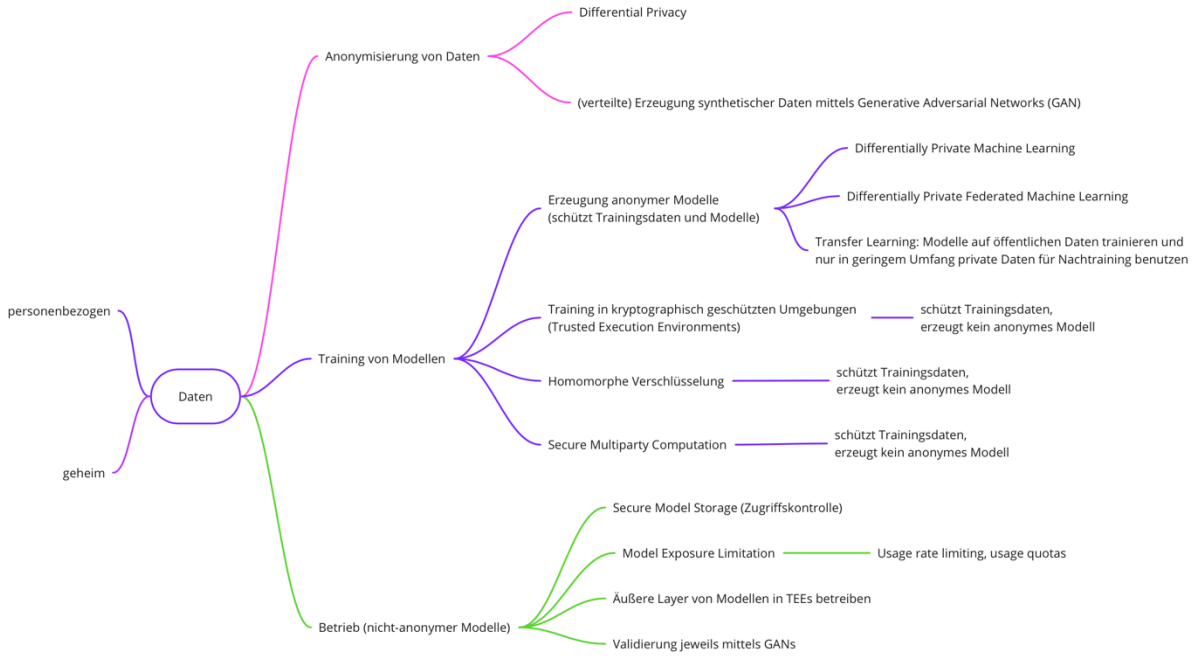

Abbildung 1: Ansätze zum Schutz von Daten und Modellen

2. Angriffe auf Machine-Learning-Modelle

Unter Privacy-Angriffen versteht man Angriffe mit dem Ziel, sensible Daten aus Modellen zu extrahieren. Einen Überblick hierzu bietet auch [1]. Die wichtigsten Angreiferziele sind *Membership Inference Attack*, *Model Inversion Attack* und *Property Inference Attack*.

Membership Inference Attacks möchten herausfinden, ob ein bestimmter Datenpunkt Teil des Trainingsdatensatzes war, der zur Erzeugung eines Modells verwendet wurde. Diese Angriffe gehen davon aus, dass aus den Ausgaben eines Modells noch Informationen über die zum Training genutzten Datenpunkte rückgewonnen werden können, z.B. dadurch, dass

Klassenwahrscheinlichkeiten in der Ausgabe für einen Datenpunkt aus den Trainingsdaten als Eingabe höher sind als für einen ungesehenen Datenpunkt. Somit wären Trainingsdaten von ungesehenen Daten unterscheidbar, was bedeutet, dass ein Angreifer für ein gegebenes Sample mit einer gewissen (Un-)Sicherheit bestimmen kann, ob dieses beim Training verwendet wurde. Kritisch sind Model Inference Attacks insbesondere dann, wenn aus der Mitgliedschaft in den Trainingsdaten eine sensible Information abgeleitet werden kann, wie beispielsweise bei medizinischen Studien. Nehmen etwa Alzheimerpatientinnen und -patienten an einer Studie teil, in der ein Modell entwickelt werden soll, um den Schwierigkeitsgrad von Gedächtnistrainings zu individualisieren, dann bedeutet ein erfolgreicher Angriff, dass für eine gegebene Person die Information rückgewonnen wird, dass diese an Alzheimer leidet.

Bei Model Inversion Attacks geht es darum, synthetische Datenpunkte oder Klassenrepräsentationen zu erzeugen. Hierfür werden, ähnlich wie bei Model Inference Attacks, die Informationen ausgenutzt, die durch die Ausgaben des Modells offengelegt werden. Angreifer versuchen somit eine Art Reverse-Engineering von plausiblen Eingaben durch Ausnutzung der Antworten des Modells durchzuführen, wodurch möglicherweise private Informationen offengelegt werden. Bei Anwendungsfällen wie der bildbasierten Personenidentifikation führt die Model Inversion Attack unmittelbar zu einem Bruch der Privatsphäre, da alle zu einer Klasse gehörigen Trainingsdaten Bilder ein- und derselben Person sind, so dass gezeigt werden konnte, dass die Rekonstruktion einer Klassenrepräsentation ein synthetisches Bild erzeugt, auf dem die betreffende Person erkennbar ist. Der potentielle Nutzen von Model Inversion Attacks hängt jedoch stark vom Anwendungsszenario ab.

Property Inference Attacks bezeichnen eine Klasse von Angriffen, bei denen ein Angreifer versucht, sensible Informationen über die Trainingsdaten insgesamt zu extrahieren, die zur Erzeugung eines Modells verwendet wurden. Durch Beobachtung des Modellverhaltens oder gezielte Abfragen kann ein Angreifer auf Eigenschaften der Trainingsdaten schließen, die nicht offengelegt werden sollen. Konkret zielen Property Inference Attacks darauf ab, bestimmte Attribute oder Eigenschaften der Trainingsdaten aufzudecken, die unter Umständen gar nicht direkt in das Training des Modells eingeflossen sind, bspw. ob ein Trainingsdatensatz zur Modellbildung für Kreditwürdigkeitsprognosen deutlich mehr Männer als Frauen oder umgekehrt enthielt und somit potentiell für ein unfaires Modellverhalten verantwortlich sein kann.

Für die Bewertung potentieller Privacy-Angriffe auf Modelle müssen jeweils szenarienabhängige Bedrohungsmodelle aufgestellt werden, die Annahmen, Fähigkeiten und Ziele von Angreifern definieren, so dass analysiert werden kann, ob eine Schwachstelle ausgenutzt werden kann. Ebenso ist die Betrachtung der Effektivität von Privacy-Angriffen auf ein gegebenes Modell wichtig, um dessen Anfälligkeit zu beurteilen. Hierzu werden in der Forschung Performanzmetriken wie beispielsweise Membership Inference Accuracy oder Reconstruction Accuracy benutzt. Liefern bekannte Angriffe bei der Übertragung auf ein neues

Szenario – etwa mittels öffentlich verfügbarer Angriffsbibliotheken – keine erfolgreichen Ergebnisse, so lässt sich auch daraus allerdings noch keine verlässliche Garantie hinsichtlich der Robustheit des betreffenden Modells gegenüber Privacy-Angriffen ableiten, da diese in der Forschung kontinuierlich verfeinert und erweitert werden.

Um Privacy-Angriffe auf Modelle zu verhindern, werden in der Literatur Mechanismen wie Differential Privacy zur Anonymisierung der Trainingsdaten, Differentially Private Machine Learning zum Trainieren anonymer Modelle, homomorphe Verschlüsselung und Secure Multiparty Computation (MPC) vorgeschlagen. Am relevantesten für die Praxis sind hierbei Methoden für Differential Privacy und Differentially Private Machine Learning die im Folgenden vorgestellt werden. Für weiterführende Informationen zu homomorpher Verschlüsselung und MPC siehe [1].

3. Training: zentral oder verteilt?

Liegen die Trainingsdaten lokal an einer zentralen Stelle vor bzw. werden diese durch eine zentrale Stelle erhoben, so muss für diese zentrale Verarbeitung personenbezogener Daten entweder ein Erlaubnistatbestand vorliegen (z.B. durch eine Einwilligung der Betroffenen) oder der Personenbezug muss frühestmöglich entfernt werden, d.h. bevor ein Zugriff auf die personenbezogenen Rohdaten möglich ist. Zur Entfernung des Personenbezugs kommt zunächst entweder eine Anonymisierung der Daten oder eine Auswertung in einer kryptographisch gesicherten Laufzeitumgebung, einer sogenannten Trusted-Execution-Environment (TEE), in Betracht. Durch eine Kombination der Ansätze können auch anonymisierte Modelle auf Rohdaten trainiert werden. Aus Datenschutzsicht ist prinzipiell eine frühestmögliche Anonymisierung zu bevorzugen.

3.1. Verteiltes Lernen

Liegen die Trainingsdaten verteilt vor, z.B. bei verschiedenen Nutzern, an verschiedenen Standorten oder in verschiedenen Organisationen, so kann das Datenschutzniveau des Trainings von Modellen dadurch gesteigert werden, dass Ansätze des verteilten bzw. föderierten Lernens (Federated Learning) genutzt werden (siehe dazu auch [1]). Hierbei verbleiben die Trainingsdaten an Ort und Stelle, wo jeweils ein lokales Modell mit den vorhandenen Daten trainiert wird. Anschließend werden die lokalen Modelle an einer zentralen Stelle zu einem generischen Gesamtmodell zusammengeführt. Somit wird der direkte Austausch der schützenswerten Daten vermieden. Wie die Forschung an verschiedenen Techniken für Privacy-Angriffe auf ML-Modelle zeigen konnte, kann man allerdings nicht davon ausgehen, dass aus einzelnen Teilmodellen oder dem generischen Gesamtmodell keine sensiblen Informationen mehr rekonstruiert werden können bzw. dass diese Modelle automatisch frei von Personenbezug wären.

Verteiltes Lernen kommt insbesondere dann in Betracht, wenn der Rechenaufwand für das Trainieren der lokalen Teilmodelle so gering ist, dass es nicht zu Leistungseinbußen bei mobilen Endgeräten kommt. Maschinelles Lernen auf Text, tabellarischen Daten oder Zeitreihen kommt somit eher für verteiltes Lernen in Betracht als lokale Bild- oder Sprachauswertung.

3.2. Vertrauenswürdige verteilte Datenerhebung bzw. verteiltes Lernen

Weiterhin stellt sich die Frage, wie sichergestellt und überprüft werden kann, dass ein ausgeliefertes Programm für verteiltes Lernen tatsächlich nur Teilmodelle an die zentrale Stelle zurückliefert und nicht etwa Rohdaten. Programme, die verteilt Trainingsdaten erheben, sind häufig Browser-Plugins oder mobile Applikationen. Moderne Software-Entwicklung folgt meist einem Continuous Integration und Development-Lebenszyklus (CI/CD). Dieser stellt sicher, dass neue Versionen eines Programms immer bestimmte Funktions- und Integrationstests durchlaufen, bevor sie ausgeliefert werden. Werden in einen solchen CI/CD-Lebenszyklus unabhängige Überprüfungen der Sicherheits- bzw. Datenschutzeigenschaften integriert - ähnliche Qualitätssicherungsmechanismen greifen in unterschiedlichen Ausprägungen in kommerziellen App-Marktplätzen von Apple, Google, Microsoft, etc. – und wird ferner mit Hilfe von Mechanismen der IT-Sicherheit bzw. Hardware-Vertrauensankern (Trusted-Computing-Technologien) sichergestellt, dass nur überprüfte Versionen des Programms ausgeliefert werden, so können sich Datengeber auf versprochene Sicherheitsgarantien verlassen. Konkret kann mittels Authentifizierungs- und Zugriffskontrollmechanismen sichergestellt werden, dass nur prüfberechtigte Entwickler Modifikationen an der Produktivversion der Programme vornehmen können. Ferner kann mittels Trusted-Computing-Mechanismen per Remote-Attestation sichergestellt werden, dass die Produktivversion nur von vertrauenswürdigen Servern kompiliert und ausgeliefert werden darf. Konzeptionell sind derartige Ansätze zum Teil in Spezifikationen für Datenräume angelegt. In [2] wird ein Konzept für Trustworthy CI/CD auf Basis von Trusted Platform Module (TPM) 2.0 als Hardwarevertrauensanker bzw. Ausgangspunkt für Integritätsmessungen vorgestellt und evaluiert. Produkte oder produktreife Bibliotheken existieren allerdings nur für einzelne Teilprobleme, so dass der Einsatz solcher Methoden in der Praxis noch nicht verbreitet stattfindet.

3.3. Anonymisierung von Trainingsdaten

Eine Anonymisierung muss möglichst zielgerichtet erfolgen, um für einen bestimmten Auswertungszweck eine möglichst hohe Datenqualität zu erhalten. Die verbreitetste Methode mit formalen Privacy-Garantien und praktisch der Stand der Technik ist Differential Privacy. Differential Privacy ermöglicht eine Aussage darüber, wie unterscheidbar zwei Datensätze voneinander sind, die sich nur um den Datenpunkt einer Person unterscheiden. Differential Privacy führt zu einem Kompromiss zwischen Privatsphäre und Nutzen. Wenn der Schutz der Privatsphäre zunimmt, kann der Nutzen oder die Genauigkeit der Analyse oder Bildverarbeitung abnehmen. Eine gängige Technik zur Herstellung von Differential Privacy ist das Hinzufügen von sorgfältig kalibriertem Rauschen zu den zu analysierenden Daten. Durch

das Hinzufügen von Rauschen wird es für einen Angreifer schwieriger, die Beiträge der einzelnen Datenpunkte zum „Informationsgehalt“ des Datensatzes zu unterscheiden. Für einen bestimmten Auswertungszweck anonymisierte bzw. verrauschte Daten sind für andere Auswertungszwecke in der Regel nicht mehr ohne zusätzliche Qualitätsverluste nutzbar. Wie schwierig sich die Etablierung von Differential Privacy darstellt, hängt auch wesentlich von der im konkreten Einzelfall zu verarbeitenden Art von Daten ab und ist beispielsweise leichter auf tabellarische Daten anzuwenden als auf Bilddaten.

Differential Privacy basiert auf dem Konzept eines Privacy-Budgets. Dieses bestimmt den Gesamtbetrag des Privacy-Verlustes, der während einer Reihe von Abfragen oder Analysen auftreten kann. Sobald das Budget ausgeschöpft ist, können keine weiteren Abfragen mehr durchgeführt werden, ohne dass die Privatsphäre beeinträchtigt wird. Techniken zur Etablierung von Differential Privacy beruhen häufig auf der Aggregation von Daten mehrerer Personen, um nützliche Erkenntnisse zu gewinnen und gleichzeitig die Privatsphäre zu schützen (Local Differential Privacy). Die Aggregation hilft dabei, die einzelnen Beiträge innerhalb des Datensatzes zu verwischen. Daneben steht das Modell eines vertrauenswürdigen Kurators (Global Differential Privacy), in dem eine vertrauenswürdige Stelle Datenpunkte von teilnehmenden Individuen oder Organisationen erhält und datenschutzfreundliche Umwandlungen durchführt, bevor sie die Daten zur Analyse freigibt. Dieser Ansatz gewährleistet den Schutz der Privatsphäre und ermöglicht gleichzeitig nützliche Analysen. Der Kurator überwacht das Privacy-Budget, das sich durch die Herausgabe einer anonymisierten Version der Daten jeweils reduziert. Durch Setzen eines nicht zu unterschreitenden Schwellwerts für das Privacy-Budget erhält man eine Kontrolle über das Risiko, dass Angreifer durch Zusammenführung verschiedener anonymisierter Versionen des Datensatzes sensible Informationen rückgewinnen können.

Differential Privacy kann auch auf maschinelle Lernmodelle angewendet werden. Techniken des Differentially Private Machine Learning sollen verhindern, dass aus Modellen sensible Informationen über Personen rückgewonnen werden können, die zu den Trainingsdaten beigetragen haben.

Zusammenfassend ist die nutzenerhaltende Anonymisierung Gegenstand intensiver Forschung. Ein Gleichgewicht zwischen Datenschutz und Nutzen zu finden, kann herausfordernd sein bzw. die Umsetzbarkeit ist stark vom jeweiligen Anwendungsfall abhängig.

3.4. Vertrauenswürdige Laufzeitumgebungen

Unter einer vertrauenswürdigen Laufzeitumgebung (Trusted Execution Environment, TEE) versteht man eine sichere und isolierte Umgebung, die innerhalb eines Computersystems arbeitet und durch Hardware wie sichere Koprozessoren unterstützt wird. Sie ist darauf ausgelegt, sensible Daten zu schützen und die Integrität und Vertraulichkeit ausgeführten Codes und verarbeiter Daten zu gewährleisten. TEEs etablieren somit eine sichere Grenze, die vertrauenswürdigen Code und Daten vom Rest des Systems trennt. Diese Isolierung

verhindert den unbefugten Zugriff und die Manipulation sensibler Daten, sogar gegen Angreifer mit administrativen Rechten auf dem System. Solche Hardware-Vertrauensanker bieten sichere Speicherung, Verschlüsselung und kryptografische Operationen. Eine gängige Implementierung von TEEs sind sichere Enklaven. Enklaven sind kleine, isolierte Ausführungsumgebungen innerhalb des Hauptprozessors, die in der Regel starke Sicherheitsgarantien für die Vertraulichkeit und Integrität von Code und Daten bieten. Verfügbare Techologien sind beispielsweise die Intel Software Guard Extensions (SGX) und Arm TrustZone.

Beim Einsatz von TEEs muss meist durch eine Überprüfung und Attestierung (Remote Attestation) des in einer Enklave ausgeführten Programmcodes sichergestellt werden, dass dieses keine schützenswerten Daten nach außer kommuniziert. Die Remote Attestation ermöglicht es einer dritten Partei, die Integrität und Sicherheit der TEE zu überprüfen bzw. einen Nachweis über die Vertrauenswürdigkeit des TEE-Zustands zu erhalten (Programmcode, Konfiguration, kryptographisches Schlüsselmaterial und ggf. Daten).

Das Ergebnis einer Berechnung in einer Enklave, also beispielsweise ein trainiertes Modell, darf entweder nicht der Geheimhaltung unterliegen oder muss ebenfalls in einer Enklave ausgeführt werden (vgl. Kapitel Betrieb). Ein Vorteil der Nutzung von TEEs für maschinelles Lernen besteht darin, dass Rohdaten verschlüsselt aufbewahrt und noch für weitere Zwecke in Enklaven ausgewertet werden können. Für diese Enklaven muss, wie bereits erwähnt, immer gelten, dass durch Überprüfung und Attestierung des darin ausgeführten Programmes sichergestellt wird, dass dieses keine schützenswerten Daten nach außen kommuniziert.

Der Einsatz von TEEs im Bereich des maschinellen Lernens ist bislang aufgrund der Einschränkungen der existierenden Technologien wie Intel SGX und ARM TrustZone noch nicht sehr verbreitet. Mit dem Aufkommen von Technologien wie AMD Secure Encrypted Virtualization (SEV), Intel Trust Domain Extensions (TDX) und insbesondere NVIDIA Confidential Computing, die abgesicherte GPU-Rechenkapazität zur Verfügung stellt, können derartige Ansätze zeitnah an Bedeutung gewinnen.

3.5. Differentially Private Machine Learning

Wie bereits erwähnt kann Differential Privacy nicht nur auf Rohdaten angewendet werden. Unter Differentially Private Machine Learning versteht man Ansätze, die maschinelle Lernverfahren mit Datenschutzgarantien verbinden. Mit Hilfe von Techniken wie *Gradient Clipping*, *Adaptive Noise Injection* oder *Secure Aggregation* kann Differential Privacy auf verschiedene maschinelle Lernalgorithmen angewendet werden, um Trainingsdaten und/oder Modellparameter zu schützen. Hierbei wird bspw. in den Lernprozess eingegriffen, um die Gewichte des Modells so zu verrauschen, dass für einen Angreifer keine Rekonstruktion der Eingangsdaten mehr möglich ist. Bei der Wahl eines geeigneten Differential Privacy-Parameters, einer Rauschverteilung und eines Privacy-Budgets muss ein Gleichgewicht zwischen Datenschutz und Nutzen für eine bestimmte Anwendung gefunden werden. Der praktische Einsatz erfordert somit eine sorgfältige Prüfung der algorithmischen

und Implementierungsdetails. Üblicherweise können geringere Performanzeinbußen erzielt werden, wenn mit Differentially Private ML ein anonymisiertes Modell trainiert wird als wenn die Trainingsdaten mit Differential Privacy anonymisiert werden. Allerdings müssen für Differentially Private ML die Rohdaten für das Training zur Verfügung gestellt werden, was aus Datenschutzsicht nicht immer möglich ist.

3.6. Differentially Private Federated Learning

Wurde ein anonymes Modell mittels Differentially Private Learning trainiert, so können für die Sicherheit gegen Angriffe zur Rekonstruktion von Eingangsdaten quantitative Garantien gegeben werden. Bei Modellen, die auf Rohdaten trainiert wurden, kann dies nicht erwartet werden bzw. gilt dies üblicherweise nicht – auch dann nicht, wenn mittels verteiltem Lernen oder in Enklaven trainiert wurde. Diese Methoden können allerdings kombiniert werden. Es existieren einsatzfähige Frameworks, bspw. Opacus von Meta und TensorFlow Privacy, mit deren Hilfe Differentially Private ML verteilt angewendet werden kann, so dass die einzelnen Teilmodelle Client-seitig anonymisiert werden, bevor sie an einer zentralen Stelle zu einem generischen, anonymen Gesamtmodell zusammengeführt werden. Somit erhält man einen Lernprozess, bei dem Rohdaten nicht ausgetauscht werden müssen, und Modelle, aus denen sich unter quantitativen Garantien keine Eingangsdaten rekonstruieren lassen.

4. Betrieb von Modellen

Nachdem die Themen Anonymität von Trainingsdaten und Modellen sowie Schutzmaßnahmen für den Trainingsprozess von Modellen behandelt wurden, stellen sich noch die Fragen, wie auf personenbezogenen Rohdaten trainierte Modelle im Betrieb geschützt werden können und wie Eingangsdaten geschützt werden können, die im operativen Betrieb in das Modell fließen.

4.1. Schutz von Modellen im Betrieb

Wie bereits dargelegt wurde, kann bei auf Rohdaten und ohne Maßnahmen zur Anonymisierung trainierten Modellen nicht davon ausgegangen werden, dass aus solchen Modellen keine sensiblen Daten rekonstruiert werden können. Allerdings können solche Angriffe durch einen Betrieb in einer kryptographisch gesicherten Laufzeitumgebung, also in einer durch eine Trusted Execution Environment bereitgestellten Enklave wirksam verhindert werden. Eine derartige Absicherung fällt in die Maßnahmenkategorie *Secure Deployment Environment*. Hierzu gehören auch Maßnahmen wie sichere Serverkonfiguration, sichere Kommunikationskanäle und Schutz vor Netzwerkangriffen. Zusätzlich können höhere Hürden gegenüber Angriffen auf Modelle im Betrieb durch Maßnahmen der Kategorie *Limiting Model Exposure* etabliert werden wie etwa die Begrenzung der Anfragerate, Zugriffskontrolle und Nutzungsquota.

Bereits trainierte Modelle können nach dem Stand der Technik nicht mehr wirksam anonymisiert werden. Verwandte Forschungsaktivitäten gehen in die Richtung, mit Hilfe von

föderiert trainierten Generative Adversarial Networks (GANs) Generatoren für qualitativ hochwertige synthetische Trainingsdaten zu erzeugen, mit deren Hilfe performante anonyme Modelle trainiert werden sollen.

4.2. Schutz von Eingangsdaten im Betrieb

Zum Schutz von Eingabedaten, die im Betrieb an ein Modell übermittelt werden, kann auch ein anonymes Modell naheliegenderweise nicht beitragen. Mit der Ausführung des Modells in einer kryptographisch geschützten Enklave kann hier am wirksamsten für Abhilfe gesorgt werden. Allerdings genügt es aus Sicht von Benutzern, die sensible Daten in das Modell einspeisen möchten, nicht, dass der Betreiber das Modell in einer Trusted Execution Environment ausführt. Der Benutzer muss zusätzlich darauf vertrauen können, dass in der Enklave das erwartete Modell ausgeführt wird und dass der in der Enklave ausgeführte Programmcode die Eingangsdaten nicht nach außen weitergibt. Wie bereits beschrieben bieten TEE-Technologien hierzu in der Regel Attestierungsprotokolle an. Bei einer Attestierung erzeugt die vertrauenswürdige Hardware der TEE einen abrufbaren Fingerabdruck des Enklavencodes. Dieser kann mit einem von unabhängiger Stelle überprüften Fingerabdruck des erwarteten Programmcodes verglichen werden. Durch diesen Abgleich ist sichergestellt, dass in der Enklave kein manipuliertes Programm ausgeführt wird, dass möglicherweise Rohdaten nach außen weitergibt. Attestierungsprotokolle bauen weiterhin üblicherweise einen vertrauenswürdigen Kommunikationskanal in die Enklave auf, über den anschließend auch die sensiblen Eingangsdaten vertraulich übertragen werden können. Ein ähnlicher Ansatz wurde im Kompetenzzentrum KARL auch in [3] entwickelt. Hier werden Konzepte entworfen, wie Methoden der erklärbaren KI mit Hilfe von Trusted Computing geschützt werden können. Diese Ansätze sind auch auf die zu erklärenden Modelle selbst übertragbar.

5. Referenzen

[1] Datenschutzkonformer Einsatz von Machine Learning mit Hilfe von Privatsphäre-wahrenden Techniken, Sven Ambrosius, Bachelorarbeit, KIT/Fraunhofer IOSB, 2023

[2] Integrity Measurement for CI/CD Build Processes Based on Trusted Platform Modules, Kerem Kara, Bachelorarbeit, KIT/Fraunhofer IOSB, 2023

[3] Absicherung von erklärbarer künstlicher Intelligenz durch TPM-basierte Attestierung, Jonas Heine, Bachelorarbeit, KIT/Fraunhofer IOSB, 2023

Dieses Forschungs- und Entwicklungsprojekt wird durch das Bundesministerium für Bildung und Forschung (BMBF) im Programm „Zukunft der Wertschöpfung – Forschung zu Produktion, Dienstleistung und Arbeit“ (Förderkennzeichen: 02L19C250) gefördert und vom Projektträger Karlsruhe (PTKA) betreut. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der Autorin / beim Autor.

www.kompetenzzentrum-karl.de

KARL
Künstliche Intelligenz
für Arbeit und Lernen

Kompetenzzentren
Arbeitsforschung

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung