MONITAUR

Endbericht | Call 18 | Projekt ID 6872

Inhalt

1 Einleitung

Die Bereitstellung von Services, die auf Machine Learning Modellen oder anderen Algorithmen basieren, birgt Risiken für deren Eigentümer.

Obwohl die Programmlogik nicht vollständig offengelegt, sondern über eines Service verwendet wird, können Angreifer versuchen, durch Interaktion (Senden von Anfragen und das beobachten der Antworten) das Verhalten des zugrunde liegenden Algorithmus aufzudecken und eine illegale Kopie zu erstellen. Auf diese Weise kann ein wertvolles Modell von Benutzern unrechtmäßig weiterverwendet oder monetarisiert werden.

MONITAUR stellt hierzu Lösungen bereit, die solche Angriffe dürch Überwachung der Interaktion entdecken können, und dem Serviceeigentümer eine Reaktion (z.b. Sperre von Benutzern) ermöglichen.
Das Projekt MONITAUR wurde im Zeitraum Jänner 2024 bis April 2025 durchgeführt und wie geplant abgeschlossen. Die Ergebnisse sind auf Github dokumentiert auch nach Projektende verfügbar.

2 Projektbeschreibung

2.1 Projektziel

Das Hauptziel dieses Projekts war die Entwicklung eines flexiblen, erweiterbaren und produktionsreifen Toolkits zur Erkennung verdächtiger Anfragen, die an APIs, die auf komplexer Programmlogik wie beispielsweise Machine-Learning-Modellen basieren, gesendet werden. Ein böswilliger Nutzer einer solchen API könnte sorgfältig erstellte Anfragen senden, um das Verhalten des zugrunde liegenden Machine-Learning-Modells zu verstehen, und es unerlaubterweise zu klonen - wodurch die geistigen Eigentumsrechte des Modelleigentümers verletzt würden. Aktuelle API-Überwachungslösungen bieten jedoch keine Analyse von Anfragen zur Identifizierung solcher Angriffe. Daher haben wir in MONITAUR das erste Überwachungstoolkit zur Erkennung verdächtiger Clients für den Schutz geistigen Eigentums entwickelt.

2.2 Zielgruppe

- **ML-Anwender und Forscher**, die daran interessiert sind, das Verhalten von Modellen während der Inferenz zu verstehen
- **Entwicklungsteams**, die Modelle in Produktivumgebungen einsetzen und effiziente Überwachungslösungen benötigen
- **Sicherheitsteams**, die Manipulation oder Modellmissbrauch verfolgen
- **Plattformentwickler**, die Detektoren in größere Überwachungs- oder Warnsysteme integrieren möchten

2.3 Projektergebnisse

Das abgeschlossene Projekt erstellte erfolgreich ein modulares, erweiterbares und auslieferbares Framework für die Überwachung von Vorhersagen von Machine Learning

Modellen. Es erzieltezwei zentrale Ergebnisse: ein wiederverwendbares Python- Paket für die Überwachung von Modelleingaben, und eine voll funktionsfähige Überwachungs-API, die sich im Prometheus Monitoring Framework (https://prometheus.io/) integrieren lässt.

Monitoring Toolkit

Der erste Teil des Projekts ist ein Python-Paket, das eine gemeinsame Schnittstelle für die Implementierung von Detektoren definiert. Diese Detektoren analysieren Modelleingaben und -ausgaben, um festzustellen, ob eine bestimmte Vorhersage verdächtig erscheint. Das Toolkit enthält mehrere integrierte Detektoren, z. B. zur Erkennung von Vorhersagen mit geringer Konfidenz, wiederholten Eingaben oder zu ähnlichen Beispielen. Jeder Detektor lässt sich leicht erweitern und an verschiedene Modalitäten (Text, Bild, Tabelle) und Anwendungsfälle anpassen. Das Paket wurde mit Blick auf Erweiterbarkeit entwickelt und verfügt über ein Registrierungssystem, eine starke interne Typisierung und ein integrationsfähiges Modul für logging. Durch diesen modularen Aufbau eignet es sich nicht nur für den direkten Einsatz in Modell-Pipelines oder Evaluierungsskripten, sondern unterstützt auch den Aufbau höherwertiger Dienste.

Das Toolkit ist unter https://github.com/darusik/monitoring_toolkit verfügbar.

Überwachungsserver

Der zweite Teil baut auf dem Toolkit auf, um dessen Funktionalität als FastAPI-Webdienst verfügbar zu machen. Dieser Server akzeptiert Erkennungsanfragen über HTTP, verarbeitet sie mit dem entsprechenden Detektor und stellt detaillierte Metriken nachverfolgbar mit Prometheus dar. Zu diesen Metriken gehören das Anfragevolumen, die Anzahl der markierten Abfragen, Konfidenzverteilungen und Fehlerprotokolle. All dies kann mit Prometheus (und optional Grafana) visualisiert werden, um einen Echtzeit-Einblick in das Modellverhalten zu erhalten. Die API kann mit Docker gestartet werden und unterstützt die sofortige Bereitstellung über Docker Compose, das sowohl die API als auch den Prometheus-Server startet. Sie enthält auch einen einfachen Befehlszeilen-Client zur Simulation von Erkennungsanfragen. Zusammen bilden diese Komponenten eine schlanke, aber leistungsstarke Überwachungslösung für Machine-Learning-Modelle. Sie ermöglicht es, ungewöhnliche Eingaben zu überwachen, verdächtige Muster in Echtzeit zu identifizieren und sich auf Bedrohungen wie Datenabweichungen, Modellmissbrauch oder bösartiges Verhalten vorzubereiten. Das gesamte System ist auf Flexibilität und Erweiterbarkeit ausgelegt: neue Detektoren können mit minimalem Aufwand eingebunden werden, die Log-Protokollierung kann an Produktionsumgebungen angepasst werden und Metriken können so gestaltet werden, dass sie benutzerdefinierte Überwachungs-Dashboards oder Alarmregeln unterstützen.

Der Code ist unter https://github.com/sbaresearch/monitaur verfügbar

2.4 Wichtigste Ergebnisse

- **Modulare Erkennungsbibliothek** mit wiederverwendbaren Basisklassen und einsatzbereiten Detektoren
- **Unterstützung für Text-, Bild- und tabellarische** Daten mit modalitätsspezifischer Hash-/Ähnlichkeitslogik
- **Konfigurierbare Detektorschwellenwerte und Logikmodi** (z. B. Entropie, Konfidenzrand)
- **Einfache Erweiterbarkeit** für benutzerdefinierter Detektoren oder zum Einbetten in Pipelines
- **FastAPI-basierter HTTP-Server**, der Erkennungsfunktionen bereitstellt
- **Prometheus-Integration** für die Erfassung von Echtzeit-Metriken
- **Metriken** umfassen Erkennungsraten, Konfidenzwerte, und Fehler
- **Docker** + Docker Compose für reproduzierbares Deployment
- **Optionaler CLI-Client und Jupyter-Notebooks** für Training und Tests
- **Gut strukturiertes, dokumentiertes und testbares** Projekt, bereit für den Einsatz in Produktion oder Forschung

3 Verlauf der Arbeitspakete

3.1 Arbeitspaket 1 - Detailplanung und Formales am Projektstart

Mit dem Abschluss dieses Arbeitspakets mit Ende Februar 2024 haben wir alle formalen Anforderungen zu Beginn des Projekts erfüllt. Wir haben einen detaillierten Projektplan erstellt, den ersten Projektblog veröffentlicht und den Antrag auf die erste Finanzierungsrate gestellt. Das Arbeitspaket wurde wie geplant abgeschlossen.

3.2 Arbeitspaket 2 - Requirements and landscape analysis

Der Fokus von Arbeitspaket 2 war die Ermittlung der Anforderungen für unser Projekt. Zunächst waren Interviews mit Experten aus der Industrie, die entweder Überwachungslösungen entwickeln oder potenzielle Nutzer von MONITAUR sein würden, geplant, um realistische Anforderungen für unser System zu erhalten. Leider erwiesen sich Details zu solchen Lösungen als äußerst vertraulich, und wir konnten keine Experten rekrutieren. Daher änderten wir den Ansatz, und fokussierten uns auf akademische Studien, die die Erfahrungen und das Interesse von Industriepraktikern an den Sicherheitsbedrohungen des maschinellen Lernens untersuchten, einschließlich der Angriffe durch Modelldiebstahl, die MONITAUR entschärfen soll. Wir veröffentlichten unseren zweiten Blog auf der Grundlage dieser Studien und nutzten sie, um die Anforderungen für MONITAUR zu formulieren. Obwohl wir den Fokusändern mussten, wurde das Arbeitspaket wie geplant bis Ende April 2024 abgeschlossen.

3.3 Arbeitspaket 3 - Collection and development of monitoring methods

Dieses Arbeitspaket widmet sich der Sammlung und Implementierung von Überwachungsmethoden. Zu Beginn des Arbeitspakets stellten wir einen deutlichen Anstieg der veröffentlichten Methoden zur Erkennung verdächtiger Kunden fest. Im

Vergleich zu unserer vorläufigen Untersuchung des Bereichs im Juli 2023 während des 18. Netidee-Call hat sich die Anzahl der Methoden verdreifacht (von 10 auf 30). Da wir uns mit allen Methoden vertraut machen wollten, nahm die erste Phase der Methodensammlung mehr Zeit in Anspruch als erwartet, woraufhin ein langwieriger Prozess der Eingrenzung der Kandidatenliste folgte. Als Folge daraus wurde die Implementierung terminlich verschoben, was zu einer Verlagerung der Arbeitslast auf die zweite Hälfte des Projekts führte. Bis Ende Dezember 2024 entwickelten wir mit dem Abschluss des Arbeitspakets ein Toolkit. Später überarbeiteten und strukturierte wir es jedoch während der Arbeit am Integrationsabschnitt von Arbeitspaket 4 um, da wir feststellten, dass unser ursprünglicher Ansatz, Lösungen aus wissenschaftlichen Publikationen zu implementieren, nicht zu der von uns gewünschten Flexibilität führte. Diese Methoden basierten beispielsweise auf einem spezifischen maschinellen Lernmodell, das für ein einziges Problem trainiert wurde und für jede andere Anwendung unbrauchbar wäre. Während der Überarbeitung haben wir anstelle der Implementierung einzelner Lösungen allgemeine Konzepte aus Forschungsarbeiten zusammengetragen und diese in mehrere generische Detektorklassen umgewandelt, die verschiedene Datenmodalitäten unterstützen und das Einbinden benutzerdefinierter Logik ermöglichen. Das endgültige Ergebnis dieses Arbeitspakets ist unter https://github.com/darusik/monitoring_toolkit verfügbar.

3.4 Arbeitspaket 4 - Monitoring Cockpit, Integration of Detectors, Testing and Deployment

Dieses Arbeitspaket konzentriert sich auf die Integration unseres Toolkits in Open-Source-Überwachungssysteme. Nach der Suche nach potenziellen Kandidaten haben wir uns für Prometheus (https://prometheus.io/) zur Erfassung und Überwachung von Metriken entschieden. Wir haben einen schlanken, erweiterbaren Überwachungsserver mit FastAPI entwickelt, der auf einem Detektor aus unserem Überwachungstoolkit basiert. Der Code ist unter https://github.com/sbaresearch/monitaur verfügbar.

Ein weiterer wichtiger Teil dieses Arbeitspakets war die Integration, das Testen und das Deployment . Wie oben erwähnt, haben wir in diesem Teil das Überwachungstoolkit überarbeitet, um mehr Flexibilität und eine bessere Anpassbarkeit zu gewährleisten. Außerdem haben wir Unit-Tests für das Toolkit hinzugefügt, es als Softwaremodul gepackt und in den Überwachungsserver integriert.

3.5 Arbeitspaket 5 - Dokumentation und Formales am Projektende

Dieses Arbeitspaket hat die abschließende Prüfung und Dokumentation des Projekts zum Inhalt. Der Projektendbericht, Zusammenfassung, Anwender:innen-Dokumentation und Entwickler:innen-Dokumentation wurden erstellt und auf die Projektwebsite bzw. github hochgeladen, bzw. dem Fördergeber übermittelt. Die Endabrechnung inkl. Originalbelege wurde übermittelt. Die Projektwebsite wurde aktualisiert und alle Ergebnisse unter Angaben der Lizenzen der Öffentlichkeit zur Verfügung gestellt.

4 Umsetzung Förderauflagen

Das Projekt hat keine Förderauflagen.

5 Liste Projektendergebnisse

1	*Projektzwischenbericht*	*CC BY-SA 4.0*	*https://www.netidee.at/monitaur*
2	*Projektendbericht*	*CC BY-SA 4.0*	*https://www.netidee.at/monitaur*
3	*Entwickler_innen-DOKUMENTATION*	*CC BY-SA 4.0*	*https://www.netidee.at/monitaur*
4	*Anwender_innen-DOKUMENTATION*	*CC BY-SA 4.0*	*https://www.netidee.at/monitaur*
5	*Veröffentlichungsfähiger Einseiter / Zusammenfassung*	*CC BY-SA 4.0*	*https://www.netidee.at/monitaur*
6	*Dokumentation Externkommunikation zur Erreichung Sichtbarkeit /Nachhaltigkeit (als Teil des Endberichtes)*	*CC BY-SA 4.0*	*https://www.netidee.at/monitaur*
7	*Monitoring framework*	*MIT*	*https://github.com/darusik/monitoring_toolkit* *https://www.netidee.at/monitaur*
8	*MONITAUR – monitoring service*	*MIT*	*https://github.com/sbaresearch/monitaur* *https://www.netidee.at/monitaur*

6 Verwertung der Projektergebnisse in der Praxis

Die Ergebnisse dieses Projekts können sowohl in der Forschung als auch in der Industrie direkt angewendet werden, um die Zuverlässigkeit und Transparenz von Machine-Learning-Modellen in der Produktion zu verbessern. Das wiederverwendbare Monitoring-Toolkit ermöglicht es Anwendern, anpassbare Erkennungsmechanismen in ihre Modell-Pipelines einzubetten, um unsichere, ungewöhnliche oder wiederholte Eingaben zu markieren– ohne dass dafür wesentliche Änderungen an bestehenden Workflows erforderlich sind. Diese Detektoren können dabei helfen, Probleme wie Modellmissbrauch, Ergebnisse mit geringer Zuverlässigkeit oder manipulierte Eingaben zu identifizieren und zu mindern.

In der praktischen Anwendung bietet der zugehörige FastAPI-basierte Überwachungsserver eine skalierbare Schnittstelle zu diesen Detektoren und lässt sich

nahtlos in Prometheus, ein weit verbreitetes Überwachungssystem, integrieren. Dies ermöglicht die Erfassung und Visualisierung von Metriken in Echtzeit und unterstützt die Entwicklung von Dashboards und Warnsystemen, die Datenanalysten und ML-Entwickler über potenzielle Modellprobleme informieren, sobald diese auftreten.

Da die Lösung containerisiert und modular aufgebaut ist, lässt sie sich mit minimalem Aufwand in bestehende MLOps-Pipelines integrieren. Sie eignet sich besonders für Teams, die nach schlanken, erweiterbaren Überwachungsfunktionen, ohne den Aufwand komplexer externer Abhängigkeiten oder plattformspezifischer Einschränkungen, suchen.

7 Öffentlichkeitsarbeit/ Vernetzung

Die Sichtbarkeit und Vernetzung von MONITAUR wurden durch verschiedene Maßnahmen erhöht, z.b. durch Präsentation auf Veranstaltungen wie dem Netidee Spring Talk, bzw. durch Präsentation der allgemeinen Problemstellung in Machine Learning in entsprechenden Meetups.

Kontakte zur Industrie aus dem Netzwerk von SBA Research wurden genutzt, um früh Feedback und Anforderungen für unsere Lösung zu bekommen, und damit auch das Projekt sowie dessen geplante Ergebnisse zu bewerben.

Mit der Finalisierung unseres Toolkits und der Monitoring Lösung, sowie einer vollständigen Dokumentation, werden wir auch verstärkt die breitere Öffentlichkeit informieren, z.b. mit Posts auf Platftformen wie X, bluesky, oder Linkedin, wobei wir hier vom Team für Öffentlichkeitsarbeit von SBA unterstützt werden.

Wir planen auch, das Projekt im Rahmen von diversen Networking Veranstaltungen aktiv zu zu präsentieren und bewerben; dazu gehören z.b. das Women4Cyber Austria Chapter, das eine hervorragende Plattform bietet, um Frauen in der Cybersecurity-Branche zu erreichen und zu fördern, sowie weiteren Veranstaltungen wie dem Cybersecurity Meetup hosted by SBA Research, oder dem Vienna Deep Learning Meetup.

8 Eigene Projektwebsite

Das Projekt MONITAUR wurde auf der Website von SBA Research vorgestellt:
https://www.sba-research.org/research/projects/monitaur/

9 Geplante Aktivitäten nach netidee-Projektende

Wir planen, das Projekt in zwei Hauptrichtungen weiterzuführen und auszubauen.

Einerseits werden wir aktuelle Forschungsergebnisse und Trends aus der Industrie im Bereich des maschinellen Lernens beobachten. Wenn neue Überwachungsstrategien vorgeschlagen werden, werden wir diese in unser

Überwachungstoolkit integrieren, um sicherzustellen, dass das Paket auf dem neuesten Stand bleibt und für reale Anwendungsfälle relevant ist.

Andererseits möchten wir eine breitere Community rund um unsere Lösung aufbauen. Dazu gehört die Bewerbung des Toolkits unter Forschern und Praktikern über Open-Source-Kanäle. Wir hoffen, Feedback zu erhalten, die Zusammenarbeit zu stärken und die Nutzer bei der Anwendung und Erweiterung unseres Überwachungsframeworks in ihren eigenen Anwendungen zu unterstützen.

Durch die Verbindung von Forschung und angewandter Praxis wollen wir die kontinuierliche Nützlichkeit und Weiterentwicklung des Projekts sicherstellen.

10 Anregungen für Weiterentwicklungen durch Dritte

Der modulare und quelloffene Charakter dieses Projekts bietet Dritten – darunter Forschern, Start-ups und MLOps-Anwendern – zahlreiche Möglichkeiten, auf den Ergebnissen aufzubauen und sie an ihre eigenen Bedürfnisse anzupassen.

Für Anwender und ML-Teams

- Nutzung vorhandener APIs oder Microservices, um Ein- und Ausgabedaten in Echtzeit zu überwachen, ohne die Modellinterna ändern zu müssen.
- Bereitstellung benutzerdefinierter Erkennungslogik für domänenspezifische Anomalien hinzu (z. B. Recht, Medizinische, Finanz, ...).
- Visualisierung des Verhaltens über die Zeit mit der Prometheus + Grafana-Integration, um fundierte Entscheidungen zur Nachschulung oder Echtzeit-Warnmeldungen zu ermöglichen.

Für Forscher

- Prototype and compare novel monitoring techniques by subclassing the BaseDetector.
- Entwicklung von Prototypen und Vergleich neuartiger Überwachungstechniken, indem die BaseDetector Klasse erweitert wird.
- Testumgebung für Explainability- oder Robustheitsmetriken.
- Quantifizierung von Datendrift oder Attacken in einer strukturierten, reproduzierbaren Umgebung.

Für Open-Source-Cumminty und Bildungseinrichtungen

- Lehrmittel für sichere KI-Einsatzpraktiken
- Bereitstellung einer erweiterbaren Codebasis, für die neue Detektoren, Metriken oder visuelle Tools entwickelt werden können

- Ermöglicht einfache Experimente über Jupyter-Notebooks oder Beispielskripte.

Empfehlungen:

Wir ermutigen Dritte, das Projekt zu forken oder zu integrieren, ihre eigenen Detektoren oder Anwendungsfälle beizusteuern und Echtzeitüberwachung als gemeinsame Herausforderung im gesamten KI-Ökosystem zu erkunden. Angesichts der wachsenden Nachfrage nach Zuverlässigkeit und Transparenz im Bereich maschinelles Lernen werden schlanke Überwachungstools wie dieses immer wichtiger – und offene Innovation kann dies weiter vorantreiben.