Fakultät Ingenieurwissenschaften
Labor für Kooperative, automatisierte Verkehrssysteme (KAV)

Verteidigungsmaßnahmen gegen Modell-Inversionsangriffe

Bachelorarbeit

von

Hannes Weber

Aschaffenburg, 24. November 2023

Autor:

Hannes Weber
Am Lindenbrunnen 17
D-97846 Partenstein

Matrikel-Nr.: 2220472
Studiengang Software Design (Bachelor)

Prüfer:

Prof. Dr.-Ing. Konrad Doll
Kooperative, automatisierte Verkehrssysteme (KAV)

Zweitprüfer:

Prof. Dr.-Ing. Ulrich Brunsmann

Technische Hochschule Aschaffenburg
Fakultät Ingenieurwissenschaften
Würzburger Straße 45
D-63743 Aschaffenburg

Ehrenwörtliche Erklärung

Hannes Weber

Am Lindenbrunnen 17
D-97846 Partenstein

Hiermit erkläre ich, dass ich die von mir vorgelegte Arbeit mit dem Thema „*Verteidigungsmaßnahmen gegen Modell-Inversionsangriffe*“ selbstständig verfasst habe, dass ich die verwendeten Quellen, Internet-Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen und Bildern –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Aschaffenburg, den 24. November 2023

Hannes Weber

Danksagung

Hiermit möchte ich mich ausdrücklich bei Herrn Prof. Dr. Konrad Doll und Herrn Prof. Dr. Ulrich Brunsmann für die Unterstützung während meiner Arbeit bedanken. Des Weiteren ...

Inhaltsverzeichnis

Abkürzungen

ADAS	*Advanced Driver Assistance Systems*

Kapitel 1

1 Einleitung

1.1 Motivation

Inmitten der fortscheitenden Digitalisierung greift man immer häufiger auf Lösungen zurück, die diverse Aufgaben in Arbeits- und Lebensbereich unterstützen, vereinfachen und sogar ersetzen. Von Medizin, in der beispielsweise Diagnoseverfahren zur frühzeitigen Erkennung von Krankheiten eingesetzt werden, bis zur Automobilindustrie, die schon jetzt auf selbstfahrende Kraftfahrzeuge setzt. Durch all die neuen Entwicklungen und Einsatzgebiete von künstlichen Intelligenzen gewinnt diese unaufhaltsam an Bedeutung.

Trotz des enormen Potenzials, das diese Technologien in ihren jeweiligen Anwendungsgebieten mit sich bringen, treten vermehrt Herausforderungen hinsichtlich Sicherheit und Robustheit der Systeme auf. Ein zentraler Aspekt ist die Notwendigkeit der Risikominimierung durch das Sicherstellen der Vertrauenswürdigkeit von KI-Systemen. Ein weiterer wichtiger Punkt in der Entwicklung dieser Systeme ist die Robustheit gegenüber Angriffen. KI-Systeme können anfällig für Manipulationen und gezielte Angriffe von außen oder durch Fehlkonfigurationen sein. Dabei spielt die Implementierung und Integration von bestimmten Sicherheitsmechanismen eine entscheidende Rolle, um die Integrität der Systeme zu gewährleisten. Insgesamt is die Sicherheit im Bereich des maschinellen Lernens von zentraler Bedeutung, wordurch das Vertrauen der Nutzer gestärkt und die breite Integration dieser Technologien in verschiedener Bereichen vorangetrieben wird. Diese Arbeit legt den Fokus auf die Herausforderung der Verdeutlichung von Sicherheits- und Robustheitsimplementierungen in Bild-Klassifikationsmodellen. In einem globalen System, in dem immer mehr Aspekte des täglichen Lebens in die 'Hände' von KI-Systemen gegeben werden, ist es umso wichtiger, diese mit Anbetracht auf Sicherheit zu implementieren, wie auch zu überwachen. Eine Unsicherheit eines Systems kann hierbei schon zu Verletzungen der Privatsphäre einzelner Personen führen. Daher wird diese Arbeit gesondert Angriffsvektoren und Bedrohungen von Klassifikationsmodellen behandeln. Darüber hinaus werden innovative Ansätze zur Bekämpfung möglicher Angriffsvektoren und Schwachstellen aufgezeigt, die sowohl Sicherheit, als auch Robustheit von neuronalen Netzen erhöhen. Insgesamt soll diese Arbeit dazu beitragen, dem Leser ein grundlegendes Verständis über Angriffe, deren Auswirkungen, wie auch Verteidigungsmaßnahmen zu liefern.

1.2 Aufgabenstellung

Hintergrund

Da KI-Systeme immer mehr Anwendung finden, werden diese zudem häufiger Ziele von Cyber-Angriffen. Es gibt viele verschiedene Angriffsvektoren auf diese Systeme, zu denen auch das Erlangen von zugrundeliegenden Informationen über Daten des Trainingsprozesses von Neuronalen Netwerken gehört. Da viele Modelle mit sensiblen Daten trainiert werden, ist es wichtig, dass Daten durch das Modell nicht an Angreifer übergeben werden. Der 'EU-AI Act' enthält unter anderem Anforderungen an KI-Systeme in Bezug auf Robustheit und Cybersicherheit. Daher bringt dieser nicht nur die Verantwortung sichere Modelle zu trainieren mit sich, sondern auch die Sicherstellung, dass genutzte Trainingsdaten privat gehalten werden. Um Modelle während der Bereitstellung abzusichern, müssen die verschiedenen Angriffsvektoren und die entsprechenden Abwehrmaßnahmen bekannt sein.

Ziel der Arbeit

Während der Thesis sollen folgende Fragen beantwortet werden:

- Welche Angriffsmöglichkeiten gibt es, um Daten von deployten Modellen zu extrahieren?
- Wie kann man sich gegen diese Attacken schützen?
- Showcase zu Verteidigungstechniken und deren Effektivität gegenüber Inversions-Angriffen.

Methodischer Ansatz

- Onboarding
- Recherche über verschiedene Attacken und Verteidigungsmöglichkeiten
- Vergleich von verschiedenen Angriffen auf unterschiedliche Modell-Architekturen
- Implementierung eines Showcases für mindestens einen Angriff auf mindestens eine Modell-Architektur:
  - Zeigen, wie ein solcher Angriff funktioniert.
  - Kann man sich gegen einen solchen Angriff verteidigen?
  - Wie effektiv sind die Verteidigungsstrategien?

1.3 Aufbau der Arbeit

Die Arbeit unterteilt sich in drei Hauptbestandteile. Dazu gehört zum einen der SStand der Technik", worin aktuelle Technologien und Modelle dargestellt werden. Hauptsächlich wurde der Wissenstand aus anderen Papern entnommen und spiegelt somit den "ResearchingTeil der Arbeit wieder. ....

Eine weitere Hauptkomponente der Arbeit ist die Implementierung eines Modells und dessen Angriffsvektoren. Mit der Implementierung der möglichen Verteidigungsmaßnahmen der Schwachstellen wird die Kehrseite aufgezeigt, mit der eine mögliche Absicherung stattfinden kann. ...

Durch die Darstellung und Auswertung der herausgefundenen Ergebnisse, geht die Arbeit zu Ende. Dabei werden die wichtigsten Erkenntnisse aus den Implementierungen und Researching-Tasks zusammengefasst, und anschaulich dargestellt. Dies soll den Vorteil mit sich bringen, dem Leser einen Überblick des neu erweorbenen und eine finale Aussicht zu bieten. ...

Der Aufbau der Arbeit lehnt sich an die Struktur von [**Bar-Shalom**] an.

Kapitel 2

2 Grundlagen

2.1 Maschinelles Lernen

2.1.1 supervised vs. unsupervised learning

2.1.2 reinforcment learning

2.2 Bilderkennung /-klassifikation

2.3 Angriffsmöglichkeiten auf Neuronale Netzwerke

Hier soll geschildert werden, welche Angriffsmöglichkeiten es gibt. Im ersten Unterpunkt soll auf Model-Inversionsangriffe eingegangen werden.

2.3.1 Modell-Inversionsangriffe

Hier soll was über Modell-Inversionsangriffe stehen (Bsp. usw.) ...

2.3.1.1 Angriffsziel

Hier soll was über Ziel von diesem Angriff geschreiben werden (Bsp. usw.) ...

2.3.1.2 Angriffsvektoren

Hier soll geschrieben werden, wie der Angriff ausgeführt werden kann (Bsp. usw.) ...

2.3.1.3 Verteidigungsstrategien

Hier sollen mögliche Verteidigungsstrategien beleuchtet werden ...

Kapitel 3

3 Stand der Technik

3.1 Forschungsergebnisse

Kapitel 4

4 Implementierung

4.1 Funktionalität des Codes

Hier soll die Funktionalität des Codes beschrieben werden ...

```
import numpy as np

def incmatrix(genl1,genl2):
  m = len(genl1)
  n = len(genl2)
  M = None #to become the incidence matrix
  VT = np.zeros((n*m,1), int)  #dummy variable

#compute the bitwise xor matrix
  M1 = bitxormatrix(genl1)
  M2 = np.triu(bitxormatrix(genl2),1)

  for i in range(m-1):
    for j in range(i+1, m):
      [r,c] = np.where(M2 == M1[i,j])
        for k in range(len(r)):
          VT[(i)*n + r[k]] = 1;
          VT[(i)*n + c[k]] = 1;
          VT[(j)*n + r[k]] = 1;
          VT[(j)*n + c[k]] = 1;

          if M is None:
            M = np.copy(VT)
          else:
            M = np.concatenate((M, VT), 1)

          VT = np.zeros((n*m,1), int)

return M
```

Listing 4.1: Python example

4.2 Code-Beschreibung

Hier soll beschrieben werden, wie der Code strukturiert ist, nach welchen Pattern gearbeitet wurde und was der Code oberflächlich macht...

Kapitel 5

Ergebnisse

5.1 Beobachtungen

Hier sollen die Beobachtungen der Verteidigungsstrategien stehen ...

Bild einfügen: Das ist ein TEstBild Hier steht ganz viel Text... Hier

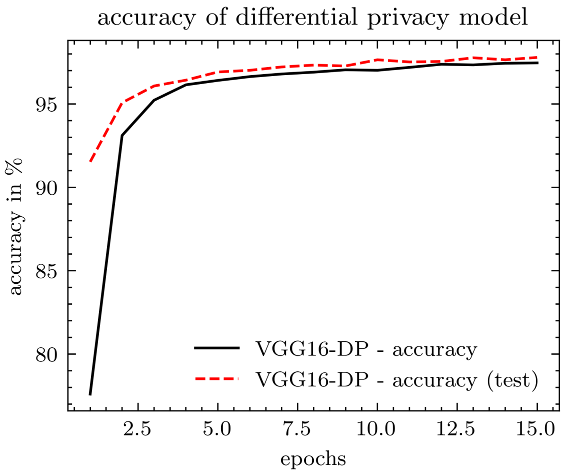

Bild 5.1: Beschreibung der Grafik

steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text... Hier steht ganz viel Text...

5.2 Rückschlüsse

Hier sollen die Rückschlüsse stehen ...

Kapitel 6

6 Zusammenfassung und Ausblick

Kapitel 7

7 Anhang

Hier sind noch LaTeX-Beispiele für die Verwendung:

...

Tabelle erzeugen:

Tabelle 7.1: Beispieltabelle

Spalte 1	Spalte 2	Spalte 3	Spalte 4
Inhalt 1	Inhalt 2	Inhalt 3	Inhalt 4
Inhalt 5	Inhalt 6	Inhalt 7	Inhalt 8

...

Quellen verlinken: Test [**Bar-Shalom**]. Test [**Goldhammer**]. Test [**WHO-2004**].

...

Kapitel verlinken: Wie in Abschnitt 5 beschrieben wird ...

...

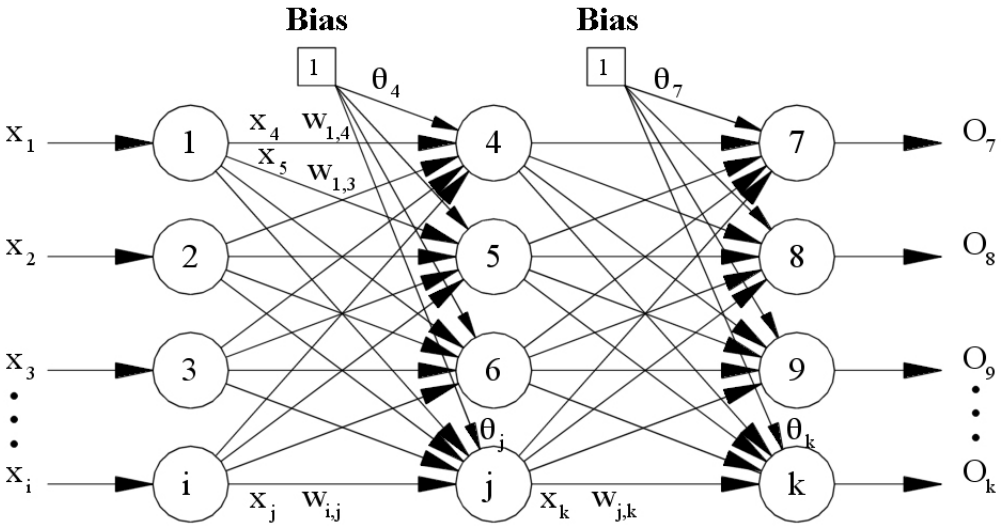

Bild 7.1: Beispielbild

Bilderverzeichnis

Tabellen

Literatur

Bücher

[1] Uwe Lorenz.
Reinforcement Learning: Aktuelle Ansätze verstehen - mit Beispielen in Java und Greenfoot. de.
Springer Berlin Heidelberg, 2020.
ISBN: 978-3-662-61650-5 978-3-662-61651-2.
DOI: 10.1007/978-3-662-61651-2.
URL: http://link.springer.com/10.1007/978-3-662-61651-2 (besucht am 04. 10. 2023).