T.I.S.P./T.P.S.S.E. Community Meeting 2025

Berlin, 04. - 05.11.2025

LLM in der Cybersicherheit

Rudolf Schreiner, ObjectSecurity OSA GmbH

TeleTrusT
Pioneers in IT security.
Bundesverband IT-Sicherheit e.V.

Agenda

- Kurze Einführung in LLM
  - Wichtig zum Verständnis der Grenzen
- Nutzungsmöglichkeiten von LLM in der Cybersecurity
- Praktische Anwendung: Cybersecurity Copilot

Disclaimer

- AI/LLM ist ein bewegliches Ziel
- Immer neue LLM Modelle und Software
  - Agenten
- Dies ist eine Einführung, keine erschöpfende Behandlung des Themas

- Sehr wichtige Aspekte werden **nicht** addressiert
  - Adversarial AI
  - Sichere Nutzung von LLM/RAG

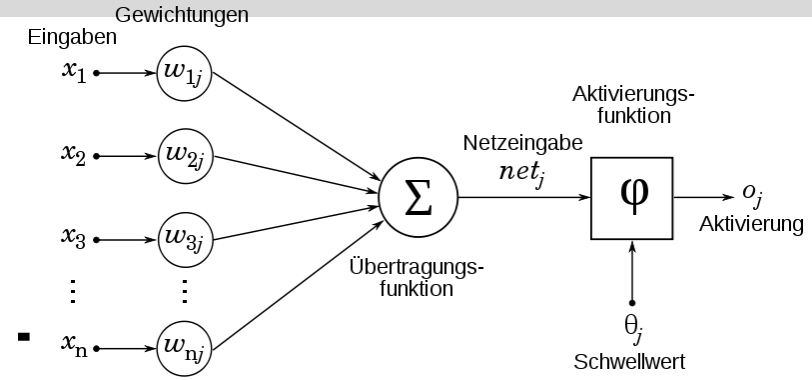
TeleTrusT
Pioneers in IT security.
Bundesverband IT-Sicherheit e.V.

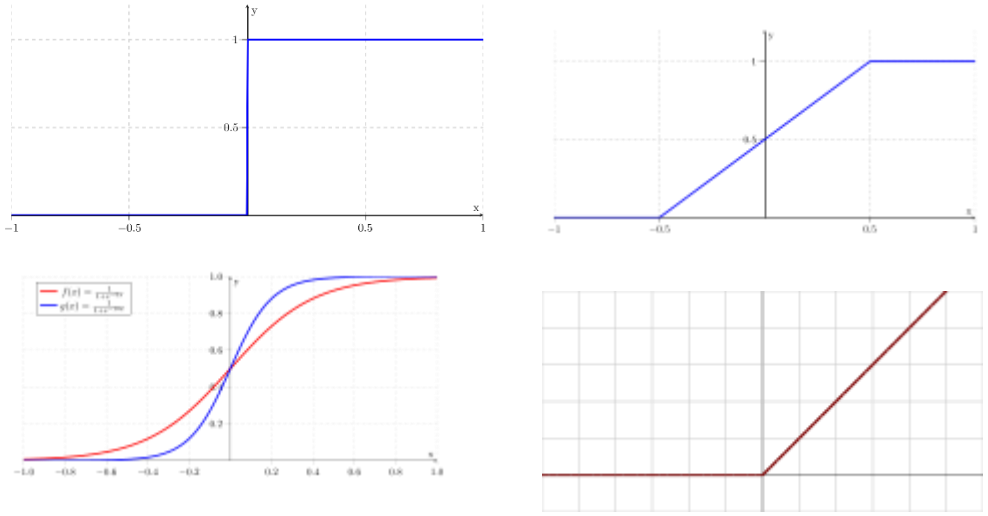
LLM: Eine Einführung

- Large Language Models (LLM) sind KI-Systeme für den Umgang mit Texten und Sprache
  - Menschliche Sprache und Text
  - Programmiersprachen und Programme
  - Strukturierte Texte, Log-Dateien
  - (Maschinencode/Binaries)
- LLM können Sprache und Texte
  - „Verstehen“
  - Generieren
  - Transformieren, also z.B. Übersetzen oder Fragen beantworten
- Multimodalität
  - Bild nach Text
  - Text nach Bild
- LLM basieren auf neuronalen Netzen

Neuronale Netzwerke: Neuron

- Neuronale Netzwerke bestehen aus künstlichen Neuronen
  - Ähnlich Nervenzellen
- Eingaben X werden mit Gewichtungen multipliziert und aufsummiert
  - Gewichtung kann positiv oder negativ sein
- Schwellenwert hinzuaddiert, dann in die Aktivierungsfunktion
- Grundlage für LLM
  - Sehr viele Neuronen
  - Spezielle Topologie: Transformer

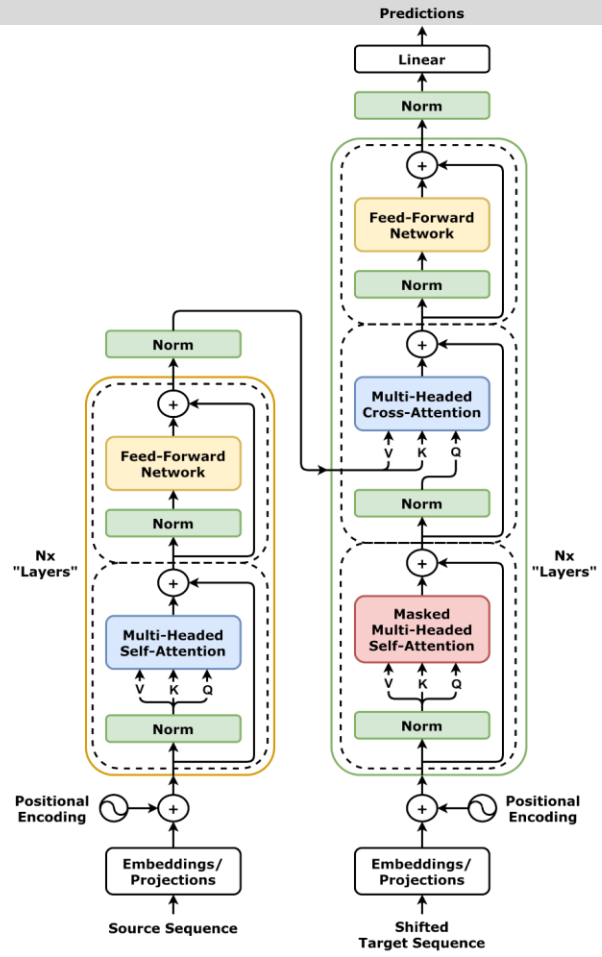




https://de.wikipedia.org/wiki/K%C3%BCnstliches_neuronales_Netz

Attention is all you need

- Wir wollen einen Satz verstehen und übersetzen:
  He lies on the bed and tells lies.
- Zweimal lies: Liegen und Lügen
- Wie kann man die unterschiedlichen Bedeutungen unterschieden?
- Wie kann man zusammengehörende Worte verbinden, einen Kontext herstellen?
  - Attention is all you need (2017, Google)
- Transformer: Details sind komplex
  - Embedding: Eingabe als Vektoren
  - Danach Folge von Encodern und Decodern
  - Dazu verwendet man grosse Menge an Trainingsdaten, also Text
- Aber wie funktioniert das intuitiv?



Wortwolken

- Beim Training zerlegen wir den Text in abstrakte Token
- Wir schauen, welche Token oft miteinander auftreten und bilden ein Art assoziativer Wortwolken, dargestellt als Vektoren in einer Vektordatenbank
- Token 1: Lies
  - Tokens: Bett, Gegenstand,Ort, Position, müde, schlafen, ausruhen, erholen...
- Token 2: Lies
  - Tokens: Sprache, Reden, Unwahrheit, Betrug,...
- Bei der Analyse von Text unterteilen wir den Text in Tokens und berechnen die Distanzen zu den Nachbar-Token
  - He lies on the bed and tells lies.
- Damit sehen wir dann, ob das gesucht Token Lügen oder Liegen bedeutet
- Im Detail ist das alles wesentlich komplexer...
- Gute Einführung bei https://www.soekia.ch/

Eigenschaften von LLM

- LLM beruhen zu 100% auf trainierten neuronalen Netzwerken
  - Keine echte Intelligenz, Moral oder Innovation
  - Vorurteile und Fehler werden reproduziert
  - Halluzinationen sind durchaus häufig
  - Nicht wirklich vertrauenswürdig
- LLM arbeiten intern mit Vektoren
  - Alle Texte und Dokumente werden in Vektoren umgewandelt
  - Wir können Texte in Vektordatenbanken speichern und verarbeiten
- Verarbeitung von exakten Detailinformation (z.B. Tabellen) oft schwierig
- LLM beruhen auf komplexer Mathematik
  - An sich deterministisch
- LLM sind SEHR gross
- Funktional können LLM nichts, was Menschen nicht auch können
  - Nur neues Werkzeug

KI als Disruption

- Wenn KI nur ein neues Werkzeug ist, warum stellt sie eigentlich alles auf den Kopf?
- Viele konzeptionell komplett neue Möglichkeiten für:
  - Erzeugung von Daten
  - Analyse von Daten
- Dazu
  - Skalierbarkeit für riesige Datenmengen
  - Automatisierung
  - Anpassung und Selbstoptimierung
  - Aus Daten kann man Aktionen ableiten
  - Integration von Agenten in komplexe Workflows

Beispiel: Phishing und Deep Fakes

- Angreifer haben das natürlich alles erkannt
- Erzeugung von gezielten und glaubwürdigen Mails
  - Stellenanzeige -> passende Bewerbung
  - Love Scamming, Enkeltrick, CEO Fraud
- Multi-Modal
  - Mail, Chat, Audio, Video
- Skalierbar
  - Bishin zur gesellschaftlichen Manipulation
- Extrem schwer zu erkennen und zu bekämpfen
  - LLM, Optimierung durch GAN
  - Awareness hilft nur begrenzt
- IMHO derzeit das grösste Risiko der KI/LLM Nutzung durch Angreifer

Anwendungen von LLM in Cybersecurity

- Sehr viele potentielle Anwendungen in der Cybersecurity
  - Threat Intelligence
  - Analyse von Schwachstellen und Malware
  - Erzeugung von Code
  - Reparatur von Programmen
  - Angriffserkennung durch Analyse von Logiles
  - Konfiguration von Systemen
  - Automatisierung von Angriffen: Pentests, Training
  - Automatisierung von Verteidigung
- Viele dieser Anwendungen sind sehr relevant
  - Aber nicht alles funktioniert (schon) gleich gut
  - Quick Win?

Cybersecurity Copilot

- Wir hätten gerne Unterstützung bei der täglichen Arbeit
  - Knowledge Management, z.B. Beantwortung von Fragen
  - Erzeugung von Texten, z.B: Sicherheitspolicies
  - Erzeugung von Code, z.B. für Tests
  - Analyse von Code, Log Files,...
  - Analyse von Systemen: Audits, Pentests,...
  - Generelle Automatisierung
- Beispiele:
  - „Was sind die Schwachstellen von X“
  - „Erzeuge ein python-Script zur Steuerung von metasploit, um Schwachstelle Y bei X auszunutzen“

Praktische Umsetzung

- Wie bauen wir also unseren Cybersecurity Copilot?
- Spezialisiert auf fachliches Domänenwissen
- Flexibles Werkzeug für Cybersecurity-Praktiker, R&D
  - Anpassbar
- Schnittstellen zur Automatisierung und Integration in bestehende Tools
  - Kommandozeile
  - Integration in Programme: python
  - REST-API
  - Agenten

Standard-LLM nicht optimal

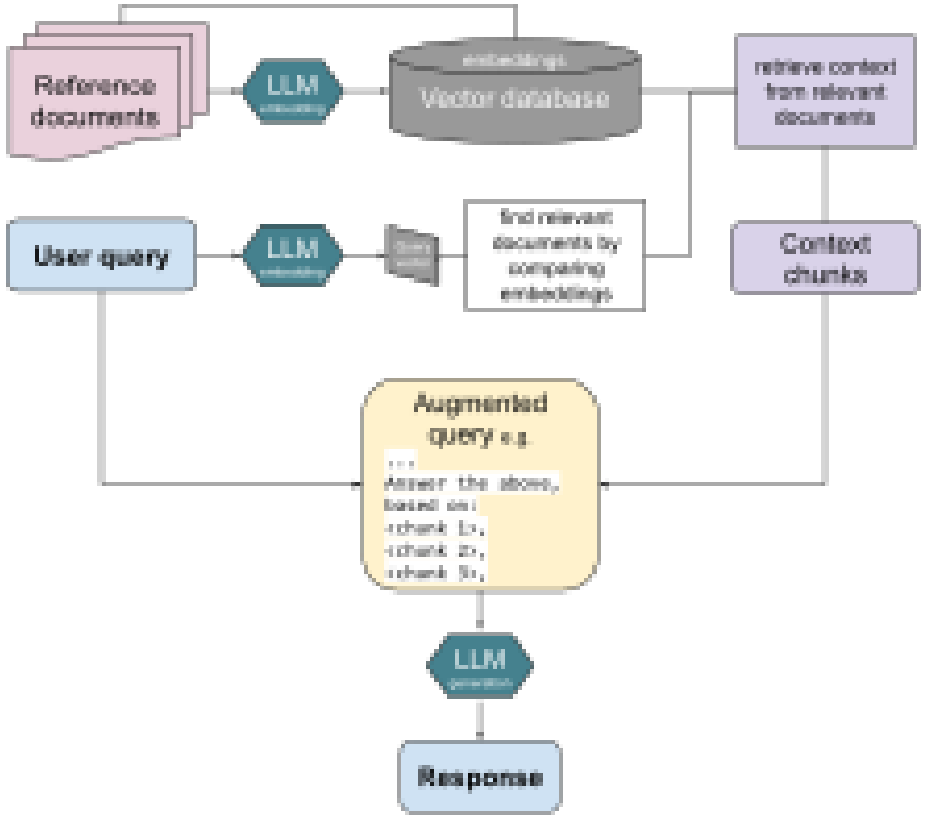
- Die bekannten LLM als Foundation Models haben im Prinzip zwei Rollen/Funktionalitäten/Aspekte
  - Grundlegendes Sprachverständnis
  - Wissen, wie Wikipedia
- Beides stammt aus dem Training und ist für konkrete Anwendungen nicht immer geeignet/optimal
  - Keine eigenen Dokumente/Texte/Daten
  - Geringe Spezialisierung
  - Aktualität und Versionierung
  - Halluzination
  - Mangelndes Vertrauen, potentielle Vergiftung
  - Grösse und Resourcenbedarf
- Optimierte LLM für Cybersecurity, z.B. RepairLlaMa
  - IMHO nur in Teilbereichen sinnvoll, z.B. Nove für Binärcode

Eigene Dokumente

- Für viele Anwendungen möchten wir eigene Informationen einbinden
  - Dokumente (pdf, Word,...)
  - Datenbanken
  - Log Files
- An sich könnte man mit diesen Dokumenten das LLM trainieren
  - Sehr aufwändig
  - Geht nicht bei Datenbanken und Logs, dynamischen Daten
- Dafür spezieller Ansatz: Retrieval Augmented Generation (RAG)
  - Trennung des Foundation Model (LLM) von den eigenen Daten
  - Also kein gemeinsames Modell, kein gemeinsamer Speicher
  - RAG-Datenbank ist viel flexibler als LLM
- Für den Nutzer aber eine integrierte Antwort aus dem LLM und den eigenen Daten

RAG

- Import eigener Dokumente in eine Vektordatenbank
- RAG besteht aus
  - Retriever: Durchsucht Datenbank nach passenden Dokumenten mit verschiendenen Techniken (Vektorsuche, Keyword, OCR,...)
  - Ranker: Bewertet und priorisiert Inhalte nach Relevanz und Nützlichkeit
  - Generator: Erzeugt Ausgabe
- RAG erzeugt einen Augmented Prompt für das LLM
- Daraus erzeugt das LLM die Ausgabe



https://en.wikipedia.org/wiki/Retrieval-augmented_generation

SLM und RAG

- LLM sind sehr leistungsfähig
  - Grosser Resourcenbedarf
  - Wissen passt nicht immer zu unseren Bedürfnissen
    - Unnützes Wissen
    - Konfikte mit RAG
- Small Language Models (SLM): Optimierung von LLM für einen bestimmten Zweck
  - Geringerer Resourcenbedarf
    - Laufen oft sehr gut auf Legacy Systemen/Notebooks und Embedded Systems
    - Keine Konflikte mit eigenen Dokumente aus RAG
    - Höhere Zuverlässigkeit
  - Kann man sehr gut mit RAG kombinieren
    - SLM liefert grundlegendes Sprachverständnis
    - Relevantes Wissen im RAG
  - Ideal für viele Anwendungen

https://en.wikipedia.org/wiki/Retrieval-augmented_generation

LLM in der Cloud

- ChatGPT, DeepSeek,...
  - Web Interface
  - App
  - API zum Einbinden in eigene Systeme
- Upload eigener Dokumente möglich, wie bei RAG
- Beste Qualität: Modelle und Performance
- ABER:
  - Hoster sieht alles und hat Zugriff auf alles
  - Einschränkungen: Nicht alles ist erlaubt
  - Legal Access durch staatliche Dienste
  - Verfügbarkeit
- Kurz: Viele Probleme in Bezug auf Sicherheit und Datenschutz

LLM lokal hosten

- Nutzung von LLM auf dem eigenen Server oder der privaten Cloud
- Llama.ccp, ollama, vllm
- Unterstützen viele LLM Modelle mit unterschiedlichen Eigenschaften
  - Qualität
  - Anwendung
  - Resourcenbedarf
  - Herkunft
  - Lizenz
- Kommandozeile für Chat
- Keine eigenen Dokumente
- API: REST und python
  - Einbindung in eigene Programme

LLM mit RAG

- REST-API des LLM
  - Meist OpenAI kompatibel
- Eigene RAG Pipeline
  - Kommandozeile
  - Direkte Integration in Python
  - API, Agenten
- Basierend auf Open Source
- Mehrere Komponenten
  - LLM Model z.B. mit ollama
  - RAG: Haystack, langchain, langflow, Llamaindex,...
  - Vektordatenbank: Chroma, Weaviate, qdrant, PostgreSQL,...
- Herausforderung: ENORM schnelle Entwickung
  - LLM-Modelle
  - Software

TeleTrusT
Pioneers in IT security.
Bundesverband IT-Sicherheit e.V.

Unser Cybersec Copilot

- Einen eigenen Copilot zu bauen ist nicht schwer
  - Ollama für LLM Modelle
  - Haystack als RAG Pipeline
  - Neo4 als Vektordatenbank
- Python-Scripts
  - Import und Chat über Kommandozeile
  - Volle Einbindung von Libraries
- Import relevanter Dokumente über Cybersecurity
  - Manuals, Papers,…
- Dient auch als Testsystem für sichere LLM/RAG
  - Zuverlässige, feingranuläre Zugriffskontrolle
- Läuft auf normalen PC mit GPU
  - Im Notfall auch auf einem Notebook

Haystack
by deepset

Copilot: Erfahrungen

- Der Copilot ist extrem nützlich und wird dauernd verwendet
- Knowledge Management, Fragen und Antworten mit Command Line
- Der Output ist generell brauchbar/hilfreich bis sehr gut
  - Kann auch direkt ausgeführt werden
- Unterschiedliche Modelle unterscheiden sich stark in Fokus und Qualität
- Integration in python ist sehr hilfreich, dadurch viele Libraries/Tools nutzbar: nmap, kismet, metasploit
- Weitere KI –Systeme können eingebunden werden: Pytorch, ART, captum
  - Anomalies Detection mit pytorch, Analyse und Reporting mit LLM/RAG

Copilot: Erfahrungen

- Output nicht innovativ
  - Versuch, eine neue SCA auch nur nachzuvollziehen scheiterte
  - Der Code wurde nach detaillierter Anweisung jedoch sehr gut erzeugt
- Output manchmal gefährlich
  - Erzeugt Pentest-Scripts, die OT Systeme crashen könnten
- Der Copilot ist nur ein Tool: A fool with a tool…

- Grosses Potential für Erweiterungen
  - Agenten, xwiki für Document Management, Knowledge Graph

Zusammenfassung

- LLM beruht auf neuronalen Netzwerken: Transformer
  - Training ist alles
  - Nicht zuverlässig
- LLM haben sehr viele Anwendungen in der Cybersicherheit
  - Von Schwachstellenanalyse bis zu kompletten Pentests mit Reporterzeugung
  - Nicht alles funktioniert (schon) gleich gut
- Quick Win: Wir bauen uns selbst ein Werkzeug für die tägliche Arbeit
  - Cybersecurity Copilot
  - Implementierung ist dank Open Source nicht schwer
  - In der Praxis sehr hilfreich