INTERNATIONALE HOCHSCHULE

Projektarbeit

Internationale Hochschule Duales Studium

Studiengang: B.Sc. Informatik

Inwieweit sind Machine-Learning-Modelle für Netzwerk-Anomalieerkennung zwischen verschiedenen Datensätzen übertragbar?

Jonas Weirauch

Matrikelnummer: 10237021

Im Wiesengrund 19, 55286 Sulzheim

Betreuende Person: Dominic Lindner

Abgabedatum: 30.09.2025

Inhaltsverzeichnis

Abkürzungsverzeichnis

AI Artificial Intelligence
DoS Denial-of-Service
IDS Intrusion Detection Systems
ML Machine Learning

1 Einleitung

1.1 Motivation und Problemstellung

Mit über 10,5 Billionen US-Dollar geschätzten jährlichen Schäden bis 2025 stellen Cyberangriffe eine der größten globalen Bedrohungen dar (World Economic Forum, 2024). Gemäß dem Global Risk Report 2024 des Weltwirtschaftsforums gehören Cyberangriffe zu den fünf bedeutendsten globalen Risiken in den nächsten Jahren, was einer Verdreifachung der finanziellen Verluste im Vergleich zu 2015 entspricht (World Economic Forum, 2024). Diese besorgniserregenden Statistiken unterstreichen die akute Notwendigkeit wirksamer Sicherheitsvorkehrungen zum Schutz kritischer Infrastrukturen (Taman, 2024).

Traditionelle signaturbasierte Intrusion Detection Systeme (IDS) erreichen zunehmend ihre Grenzen bei der Erkennung neuartiger Zero-Day-Exploits und unbekannter Angriffsmuster (Belavagi & Muniyal, 2016; Ring et al., 2019). Diese Systeme können lediglich bekannte Signaturen identifizieren und versagen bei der Detektion innovativer Bedrohungen. Gleichzeitig führen die steigende Vernetzung und Digitalisierung zu einer kontinuierlichen Zunahme der Angriffsvektoren und einer erhöhten Komplexität der Netzwerkumgebungen.

Machine Learning (ML) bietet das Potenzial, diese Limitationen zu überwinden und auch bisher unbekannte Angriffsmuster aufzudecken (Vinayakumar et al., 2019). Dennoch ist die tatsächliche Wirksamkeit verschiedener ML-Modelle in heterogenen Netzwerken noch nicht vollständig geklärt. Ein kritisches Problem stellt dabei die Generalisierungsfähigkeit dar: Während Modelle auf spezifischen Trainingsdaten exzellente Leistungen erzielen, zeigen sie oft dramatische Leistungseinbußen beim Transfer auf neue Netzwerkumgebungen oder unterschiedliche Datensätze (Ring et al., 2019).

1.2 Forschungsfrage und Zielsetzung

Diese Arbeit untersucht systematisch die Generalisierungsfähigkeit von zwölf ML-Modellen über zwei fundamental unterschiedliche Netzwerk-Datensätze hinweg. Die zentrale Forschungsfrage lautet:

„Inwieweit sind Machine-Learning-Modelle für Netzwerk-Anomalieerkennung zwischen verschiedenen Datensätzen übertragbar?"

Die Untersuchung fokussiert sich auf die Cross-Dataset-Transferierbarkeit zwischen dem NSL-KDD-Datensatz (Canadian Institute for Cybersecurity, 2024b) (simulierter Netzwerkverkehr von 1998 mit 125.973 Trainingsdatensätzen) und dem CIC-IDS-2017-Datensatz (Canadian Institute for Cybersecurity, 2024a; Sharafaldin et al., 2018) (realistischer Netzwerkverkehr mit 2,8 Millionen Datenpunkten aus einer fünftägigen Netzwerkumgebung). Diese Datensätze unterscheiden sich fundamental in ihrer Datenverteilung, Merkmalsdimensionalität und den abgebildeten Angriffsszenarien (Mourouzis & Avgousti, 2021).

Die konkreten Forschungsziele umfassen:

- **Vergleichende Evaluation**: Systematische Bewertung von Baseline-Modellen (Random Forest, Decision Tree, k-NN) und Advanced-Modellen (XGBoost, LightGBM, Neural Networks) hinsichtlich ihrer Intra-Dataset-Performance und Cross-Dataset-Robustheit.

- **Cross-Dataset-Transferierbarkeit**: Quantifizierung der Generalisierungslücken beim Transfer zwischen NSL-KDD und CIC-IDS-2017 sowie Identifikation der robustesten Algorithmen für heterogene Netzwerkumgebungen.
- **Praktische Effizienzbetrachtung**: Analyse des Trade-offs zwischen Erkennungsleistung und computational Effizienz durch systematische Messung von Trainings- und Inferenzzeiten zur Bewertung der Praktikabilität in Echtzeit-Systemen.

Die Ergebnisse sollen konkrete Handlungsempfehlungen für die effektive Anwendung von ML-Modellen in verschiedenen Netzwerkszenarien liefern und zur aktuellen Forschungslandschaft der adaptiven Anomalieerkennung beitragen.

1.3 Aufbau der Arbeit

Die Arbeit gliedert sich in vier aufeinander aufbauende Hauptteile. Zunächst werden in den *theoretischen Grundlagen* die konzeptionellen Fundamente der Netzwerk-Anomalieerkennung etabliert. Dabei erfolgt eine systematische Einordnung signaturbasierter versus anomaliebasierter Detektionsansätze sowie eine Taxonomie der eingesetzten Machine-Learning-Verfahren – von traditionellen Algorithmen wie Random Forest über moderne Ensemble-Methoden bis hin zu neuronalen Netzen (McHugh, 2000; Vinayakumar et al., 2019).

Im *methodischen Teil* wird das dreistufige Evaluationsframework vorgestellt, das Within-Dataset-Validation, Cross-Dataset-Transfer und Feature-Harmonisierung systematisch kombiniert. Besondere Berücksichtigung finden dabei die Herausforderungen der Datenvorverarbeitung und die Behandlung von Klassenimbalance in heterogenen Netzwerkumgebungen (Gharib et al., 2016).

Die *empirische Analyse* präsentiert die Ergebnisse der umfassenden Modellvergleiche zwischen NSL-KDD und CIC-IDS-2017. Neben klassischen Performance-Metriken werden neuartige Transfer-Kennzahlen wie Generalization Gap und Transfer Ratio eingeführt, um die Cross-Dataset-Robustheit quantitativ zu bewerten. Feature-Importance-Analysen decken die prädiktiven Schlüsselvariablen auf und charakterisieren deren datensatzspezifische Eigenschaften.

Abschließend werden in der *Diskussion* die praktischen Implikationen für IDS-Deployments erörtert. Die Erkenntnisse münden in konkrete Handlungsempfehlungen für die Modellauswahl sowie einen Ausblick auf zukünftige Forschungsrichtungen in Transfer Learning und Explainable AI für Cybersicherheitsanwendungen. Der wissenschaftliche Beitrag liegt in der erstmaligen systematischen Cross-Dataset-Evaluation von zwölf ML-Modellen unter realistischen Transferbedingungen und der empirischen Quantifizierung von Generalisierungslücken zwischen historischen und modernen IDS-Benchmarks.

2 Theoretische Fundierung

2.1 Grundlagen der Netzwerk-Anomalieerkennung und Intrusion Detection Systems

Die Erkennung von Anomalien im Netzwerkverkehr stellt einen fundamentalen Baustein moderner Cybersicherheitsarchitekturen dar. Intrusion Detection Systems (IDS) fungieren als Frühwarnsysteme, die darauf ausgelegt sind, ungewöhnliche Muster im Netzwerkverkehr zu identifizieren, welche auf potenzielle Sicherheitsbedrohungen hindeuten könnten (Ring et al., 2019). Diese Systeme operieren kontinuierlich im Hintergrund und analysieren den gesamten Datenfluss einer Netzwerkinfrastruktur, um Angriffe wie Denial-of-Service (DoS), unbefugtes Eindringen, Datenexfiltration oder Malware-Aktivitäten zu erkennen (Vinayakumar et al., 2019).

Die theoretische Grundlage der Anomalieerkennung basiert auf der systematischen Unterscheidung zwischen normalem und abnormalem Netzwerkverhalten. Dabei lassen sich drei fundamentale Kategorien von Anomalien differenzieren (Ring et al., 2019). **Punktuelle Anomalien** bezeichnen einzelne Datenpunkte, die signifikant von der erwarteten Normalverteilung abweichen, wie beispielsweise ungewöhnlich hohe Banbreitennutzung durch einzelne Verbindungen. **Kontextuelle Anomalien** sind Datenpunkte, die nur unter Berücksichtigung ihres spezifischen Kontexts als anormal klassifiziert werden können. Ein hoher Datenverkehr während Nachtstunden könnte kontextuell anomal sein, obwohl derselbe Verkehr während der Geschäftszeiten normal erscheint. **Kollektive Anomalien** beziehen sich auf Gruppen von Datenpunkten, die gemeinsam ein ungewöhnliches Verhalten zeigen, obwohl einzelne Werte innerhalb normaler Parameter liegen könnten, wie etwa koordinierte Botnet-Aktivitäten (Ring et al., 2019).

Die praktische Implementierung von IDS erfordert jedoch mehr als nur die technische Fähigkeit zur Mustererkennung. Moderne Netzwerkumgebungen sind durch hohe Dynamik, heterogene Infrastrukturen und kontinuierliche evolvierende Bedrohungslandschaften charakterisiert. (Gharib et al., 2016). Dies führt zu dem Phänomen des **Concept Drift**, bei dem sich die statistische Verteilung der Netzwerkdaten über die Zeit verändert, was die Anpassungsfähigkeit und Generalisierungsfähigkeit der eingesetzten Detektionssysteme vor erhebliche Herausforderungen stellt (Ring et al., 2019).

2.2 Traditionelle versus Machine Learning-basierte Detektionsansätze

2.3 Machine Learning-Taxonomie für Anomalieerkennung

2.4 Transfer Learning und Cross-Dataset-Generalisierung

3 Methodik

Design, Daten, Preprocessing, Metriken, Validierung.

3.1 Daten

Kurzbeschreibung der Datensätze.

3.2 Modelle und Hyperparameter

Tabellenbeispiel mit Quellenangabe (10 pt):

Parameter	Wert A	Wert B
Lernrate	0,001	0,01
Batchgröße	64	64

Tab. 1: Beispielhafte Hyperparameter.

Eigene Darstellung.

4 Ergebnisse

Beispielabbildung mit Titel und Quelle (10 pt):

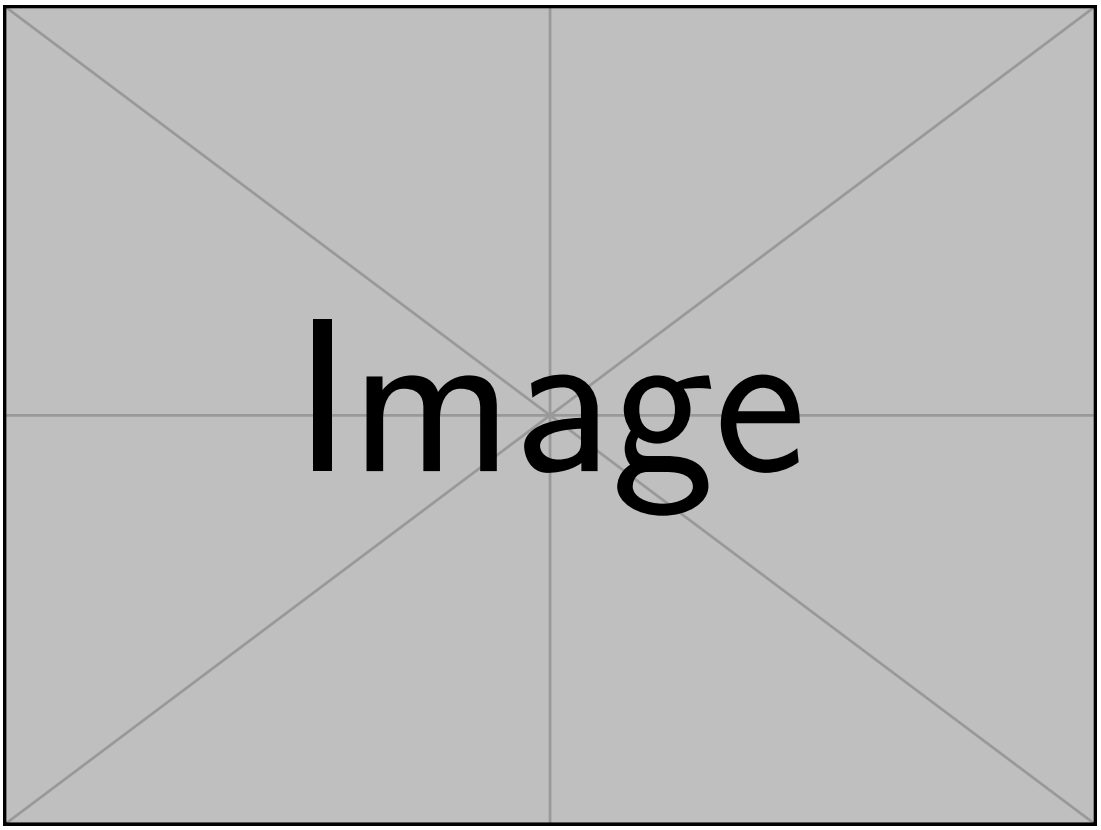

Abb. 1: Schematische Lernkurven.

Eigene Darstellung (Platzhalter).

5 Diskussion

Ergebnisse interpretieren, Limitationen, Implikationen.

6 Fazit

Zentrale Punkte, Ausblick, Handlungsempfehlungen.

Literaturverzeichnis

Belavagi, M. C., & Muniyal, B. (2016). Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection. *Procedia Computer Science*, *89*, 117–123. DOI: 10.1016/j.procs.2016.06.016.

Canadian Institute for Cybersecurity. (2024a). IDS 2017 | Datasets | Research | Canadian Institute for Cybersecurity | UNB. Verfügbar 29. März 2025 unter https://www.unb.ca/cic/datasets/ids-2017.html

Canadian Institute for Cybersecurity. (2024b). NSL-KDD | Datasets | Research | Canadian Institute for Cybersecurity | UNB. Verfügbar 29. März 2025 unter https://www.unb.ca/cic/datasets/nsl.html

Gharib, A., Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2016). An Evaluation Framework for Intrusion Detection Dataset. *2016 International Conference on Information Science and Security (ICISS)*, 1–6. DOI: 10.1109/ICISSEC.2016.7885840.

McHugh, J. (2000). Testing Intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory. *ACM Transactions on Information and System Security*, *3*(4), 262–294. DOI: 10.1145/382912.382923.

Mourouzis, T., & Avgousti, A. (2021). Intrusion Detection with Machine Learning Using Open-Sourced Datasets. DOI: 10.48550/ARXIV.2107.12621.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A Survey of Network-based Intrusion Detection Data Sets. *Computers & Security*, *86*, 147–167. DOI: 10.1016/j.cose.2019.06.005.

Sharafaldin, I., Habibi Lashkari, A., & Ghorbani, A. A. (2018). Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization, 108–116. DOI: 10.5220/0006639801080116.

Taman, D. (2024). Impacts of Financial Cybercrime on Institutions and Companies. *Arab Journal of Arts and Humanities*, *8*(30), 477–488. DOI: 10.21608/ajahs.2024.341707.

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set, 1–6. DOI: 10.1109/CISDA.2009.5356528.

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System. *IEEE Access*, *7*, 41525–41550. DOI: 10.1109/ACCESS.2019.2895334.

World Economic Forum. (2024). *Global Risks Report 2024*. World Economic Forum. Verfügbar 29. März 2025 unter https://www.weforum.org/publications/global-risks-report-2024/

Anhangsverzeichnis

A Zusatzabbildungen

(Optionaler Inhalt des Anhangs.)

B Pseudocode

(Optionaler Inhalt des Anhangs.)

Nützliche LaTeX-Referenz

Zitieren nach APA 7 (biblatex-apa)

Indirektes Zitat: `\parencite{Goodfellow2016}` → (Goodfellow et al., 2016)
Mit Seitenzahl: `\parencite[S.~123]{Bishop2006}`
Direktzitat ≤40 Wörter: „…“ `\parencite[S.~45]{Hastie2009}`
Blockzitat ≥40 Wörter:

```
\begin{blockzitat}
    Langes Zitat ohne Anführungszeichen …
\end{blockzitat}
```

Abbildungen

```
\begin{figure}[h]
    \centering
    \includegraphics[width=0.85\textwidth]{pfad/zur/datei}
    \caption{Titel der Abbildung.}
    \source{Quelle: Eigene Darstellung / Autor, Jahr, S.~xx.}
    \label{fig:beispiel}
\end{figure}
```

Querverweis: „siehe Abb. `\ref{fig:beispiel}`“.

Tabellen

```
\begin{table}[h]
    \centering
    \begin{tabular}{lcc}
        \toprule
        \textbf{Variable} & \textbf{Gruppe A} & \textbf{Gruppe B} \\
        \midrule
        x & 1{,}23 & 4{,}56 \\
        \bottomrule
    \end{tabular}
    \caption{Titel der Tabelle.}
    \source{Quelle: Eigene Darstellung.}
    \label{tab:beispiel}
\end{table}
```

Querverweis: „siehe Tab. `\ref{tab:beispiel}`“.

Gleichungen

Einzeln:

```
\begin{equation}
    E = mc^2
\end{equation}
```

Mehrzeilig (nummeriert):

```
\begin{align}
    \hat{R}(\theta) &= \frac{1}{N}\sum_{i=1}^{N}\ell(y_i,f_\theta(x_i)) + \lambda\lVer
    \ell(y,\hat{y}) &= -\big[y\log\hat{y} + (1-y)\log(1-\hat{y})\big].
\end{align}
```

Listen

```
\begin{itemize}
    \item Punkt A
    \item Punkt B
\end{itemize}

\begin{enumerate}
    \item Erstens
    \item Zweitens
\end{enumerate}
```

Fußnoten

```
Text\footnote{Inhalt der Fußnote in 10 pt.}
```

Einheiten und Zahlen (siunitx)

`\SI{12,5}{\kilo\meter\per\hour}` → 12.5 km h^{-1}
`\num{12345,678}` → 12 345.678

Quellen in Abbildungen/Tabellen

Direkt unter `\caption` einfügen: `\source{Quelle: …}` (10 pt).

Platzhalter & Blindtext

Platzhalterbild: `\includegraphics{example-image}` (aus Paket `mwe`).
Kurzer Blindtext:

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
```

Bibliografie-Einträge (BibTeX mit Biber)

Wichtige Eintragstypen:

```
@book{key,
  author = {Nachname, Vorname},
  year = {2023},
  title = {Titel des Buches},
  subtitle = {Untertitel (optional)},
  publisher = {Verlag},
  address = {Ort},
  edition = {2}, % nur bei 2. Auflage oder höher
  doi = {10.1000/xyz}
}

@article{key,
  author = {Nachname, Vorname and Zweiter, Autor},
  year = {2023},
  title = {Titel des Artikels},
  journaltitle = {Name der Zeitschrift},
  volume = {42},
  number = {3},
  pages = {123--145},
  doi = {10.1000/xyz}
}

@online{key,
  author = {Nachname, Vorname},
  year = {2023},
  title = {Titel der Webseite},
  url = {https://example.com},
  urldate = {2024-01-15}
}
```

Biber-spezifische Felder:

- `journaltitle` statt `journal` (APA-konform)

- `location` statt `address` (moderne biblatex-Syntax)
- `date` statt `year` für komplexere Datumsangaben

Code-Beispiele in LaTeX

Einfacher Python-Code:

```python
def fibonacci(n: int) -> list[int]:
"""Berechnet die ersten n Fibonacci-Zahlen."""
seq = [0, 1]
for i in range(2, n):
seq.append(seq[-1] + seq[-2])
return seq[:n]


if __name__ == "__main__":
print("Fibonacci(10):", fibonacci(10))
```

Listing 1: Fibonacci-Beispiel

Ausgabe:

```
Fibonacci(10): [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
```

—

Inline-Nutzung (LaTeX-Syntax wörtlich):

```
\begin{lstlisting}[language=Python, caption={Minimalbeispiel}, label={lst:mini}]
    def foo(x):
    return x**2
\end{lstlisting}
```

Tatsächliches Listing (ausführbarer Code):

```python
def foo(x):
return x**2
```

Listing 2: Minimalbeispiel

Ausgabe:

```
>>> foo(5)
25
```

—

Code aus Datei einbinden: `\lstinputlisting[language=Python, caption={Script X}, label={lst:scriptx}]{path/to/script.py}`

Erweiterte LaTeX-Tipps

Mathematik:

- Inline-Mathe: `$E = mc^2$` → $E = mc^2$
- Display-Mathe: `\[E = mc^2\]` (unnummeriert)
- Nummerierte Gleichung: `\begin{equation}...\end{equation}`
- Griechische Buchstaben: `\alpha, \beta, \gamma` → α, β, γ

Querverweise:

- Label setzen: `\label{fig:beispiel}`
- Verweis: `\ref{fig:beispiel}` oder `\autoref{fig:beispiel}`
- Seitenverweis: `\pageref{fig:beispiel}`

Typografie:

- Geschützte Leerzeichen: `Abb.~\ref{fig:1}`
- Anführungszeichen: `\enquote{Text}` (sprachabhängig)
- Gedankenstrich: `--` (Bindestrich), `---` (Gedankenstrich)
- Auslassungspunkte: `\ldots` → …

Häufige Probleme und Lösungen:

- Biber-Cache löschen: `biber --cache-clear`
- Umlaute: Verwende `fontspec` mit LuaLaTeX/XeLaTeX
- Lange URLs: `\url{...}` oder `\href{url}{Text}`
- Overfull hbox: `\sloppy` oder manuelle Zeilenumbrüche

Kompilierreihenfolge mit Biber

Standard: LuaLaTeX → Biber → LuaLaTeX → LuaLaTeX

VS Code/Automatisierung:

- LaTeX Workshop Extension konfigurieren
- `latexmkrc` für automatische Biber-Ausführung
- Overleaf nutzt automatisch die richtige Reihenfolge