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

Abbildungsverzeichnis

Abkürzungsverzeichnis

AI	Artificial Intelligence
AUC	Area Under the Curve
CIC-IDS-2017	Canadian Institute for Cybersecurity Intrusion Detection System 2017
DoS	Denial-of-Service
EFB	Exclusive Feature Bundling
FPR	False Positive Rate
GOSS	Gradient-based One-Side Sampling
HIDS	Host-based Intrusion Detection Systems
IDS	Intrusion Detection Systems
k-NN	k-Nearest Neighbors
ML	Machine Learning
MLP	Multi-Layer Perceptron
NIDS	Network-based Intrusion Detection Systems
NSL-KDD	Network Security Laboratory - Knowledge Discovery and Data Mining
ROC	Receiver Operating Characteristic
SVM	Support Vector Machines
TPR	True Positive Rate

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). 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). Machine Learning (ML) bietet das Potenzial, diese Limitationen zu überwinden, jedoch 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 (Ring et al., 2019).

Forschungslücke: Bisherige Studien konzentrieren sich primär auf Within-Dataset-Evaluationen und vernachlässigen die praktisch relevante Frage der Cross-Dataset-Transferierbarkeit (Mourouzis & Avgousti, 2021). Die systematische Bewertung der Generalisierungsfähigkeit zwischen fundamental verschiedenen Netzwerk-Datensätzen, insbesondere zwischen historischen Benchmarks wie NSL-KDD und modernen Datensätzen wie CIC-IDS-2017, bleibt eine unzureichend erforschte, aber praxiskritische Herausforderung.

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 NSL-KDD (1998, 41 Features) und CIC-IDS-2017 (2017, 79 Features), die sich fundamental in Datenverteilung, Merkmalsdimensionalität und Angriffsszenarien unterscheiden (Mourouzis & Avgousti, 2021).

Die konkreten Forschungsziele umfassen erstens die **systematische Cross-Dataset-Evaluation** durch bidirektionale Transferanalyse mit zwölf ML-Algorithmen von Baseline-Modellen (Random Forest, Decision Tree, k-NN, Logistic Regression, Naive Bayes, Linear SVM) bis zu Advanced-Modellen (XGBoost, LightGBM, Gradient Boosting, Extra Trees, MLP, Voting Classifier). Zweitens erfolgt die **Entwicklung neuartiger Transfer-Metriken** durch Einführung von Generalization Gap, Transfer Ratio und Relative Performance Drop als quantitative Maße für Cross-Dataset-Robustheit. Drittens wird eine **Feature-Space-Harmonisierung** zur Überbrückung der Dimensionalitätslücke (41 vs. 79 Dimensionen) implementiert. Viertens zielt die Arbeit auf **praktische Deployment-Guidance** durch Identifikation der transferrobustesten Algorithmen für heterogene Netzwerkumgebungen ab.

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, einschließlich einer Taxonomie der eingesetzten Machine-Learning-Verfahren (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 (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.

Abschließend werden in der *Diskussion* die praktischen Implikationen für IDS-Deployments erörtert. Der wissenschaftliche Beitrag liegt in der erstmaligen systematischen Cross-Dataset-Evaluation von zwölf ML-Modellen unter realistischen Transferbedingungen sowie der Entwicklung neuartiger Transfer-Metriken für ML-basierte Cybersecurity-Systeme.

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).

Architektonische Klassifikation von IDS erfolgt primär nach zwei Dimensionen: dem Einsatzort und der Detektionsmethodik (Ring et al., 2019). **Network-based IDS (NIDS)** überwachen den Netzwerkverkehr an strategischen Punkten und analysieren Pakete in Echtzeit, während **Host-based IDS (HIDS)** direkt auf einzelnen Systemen implementiert werden und Systemlogs, Dateizugriffe und Prozessaktivitäten überwachen. **Hybrid-Systeme** kombinieren beide Ansätze zur Maximierung der Abdeckung und Minimierung blinder Flecken (Gharib et al., 2016). Die Wahl der Architektur beeinflusst fundamental die verfügbaren Feature-Sets und damit die Anwendbarkeit verschiedener ML-Algorithmen.

Deployment-Modi unterscheiden zwischen passiver Überwachung durch Mirroring von Netzwerktraffic und aktiver Inline-Implementierung mit direkter Paketfilterung. Passive Systeme bieten den Vorteil der Latenz-neutralen Überwachung, während Inline-Systeme proaktive Threat-Mitigation ermöglichen, jedoch Durchsatz-Limitationen unterliegen (Vinayakumar et al., 2019). Diese architektonischen Entscheidungen determinieren die verfügbaren Datencharakteristika und beeinflussen die Generalisierbarkeit trainierter Modelle zwischen verschiedenen Netzwerkumgebungen.

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 Bandbreitennutzung 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 kontinuierlich 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

Die Evolution der Anomalieerkennungstechnologien lässt sich in zwei fundamentale Paradigmen unterteilen: signaturbasierte und anomaliebasierte Verfahren, wobei letztere zunehmend durch Machine Learning-Ansätze implementiert werden (Belavagi & Muniyal, 2016; Ring et al., 2019).

Signaturbasierte Systeme operieren nach dem Prinzip des Musterabgleichs und vergleichen den aktuellen Netzwerkverkehr mit einer Datenbank bekannter Angriffssignaturen (Ring et al., 2019). Diese Systeme zeichnen sich durch hohe Präzision bei der Erkennung bereits katalogisierter Bedrohungen aus und generieren typischerweise niedrige False-Positive-Raten. Die fundamentale Limitation signaturbasierter Ansätze liegt jedoch in ihrer Reaktivität: Sie können ausschließlich Angriffe identifizieren, deren Signaturen bereits in der Datenbank hinterlegt sind (Vinayakumar et al., 2019). Diese Eigenschaft macht sie anfällig für Zero-Day-Exploits, polymorphe Malware und neuartige Angriffstechniken, die noch nicht in den Signaturdatenbanken erfasst sind.

Anomaliebasierte Systeme verfolgen einen proaktiven Ansatz, indem sie zunächst ein statistisches Modell des normalen Netzwerkverhaltens etablieren und anschließend Abweichungen von diesem Baseline-Verhalten als potenzielle Bedrohungen klassifizieren (Ring et al., 2019). Der entscheidende Vorteil dieses Paradigmas liegt in der theoretischen Fähigkeit zur Detektion unbekannter Angriffsmuster und Zero-Day-Exploits (Vinayakumar et al., 2019). Allerdings erfordert die praktische Umsetzung eine präzise Modellierung des Normalverhaltens sowie die Definition geeigneter Schwellenwerte zur Minimierung von False-Positive-Meldungen.

Machine Learning-basierte Ansätze haben das Potenzial, die Limitationen beider traditioneller Paradigmen zu überwinden. Überwachte Lernverfahren können komplexe, nichtlineare Beziehungen zwischen Netzwerkfeatures und Angriffskategorien erlernen, während unüberwachte Methoden in der Lage sind, neuartige Anomaliemuster ohne vorherige Kennzeichnung zu identifizieren (Vinayakumar et al., 2019). Die Integration von Deep Learning-Techniken ermöglicht zudem die automatische Feature-Extraction aus hochdimensionalen Netzwerkdaten, wodurch manuell entwickelte Heuristiken obsolet werden (Goodfellow et al., 2016).

2.3 Machine Learning-Taxonomie für Anomalieerkennung

Die systematische Evaluation von ML-Verfahren in der Netzwerk-Anomalieerkennung erfordert eine strukturierte Kategorisierung nach Komplexität und methodischen Ansätzen. Diese Arbeit implementiert eine zweigeteilte Evaluationsstrategie mit sechs Baseline-Modellen und sechs Advanced-Modellen, um sowohl etablierte als auch moderne Verfahren zu bewerten (Vinayakumar et al., 2019).

Baseline-Modelle repräsentieren etablierte, interpretierbare Algorithmen mit moderater Komplexität und geringen computational Anforderungen. **Random Forest** implementiert Ensemble-Learning durch Bootstrap Aggregating (Bagging) von Entscheidungsbäumen und reduziert Overfitting durch Diversifikation (Hastie et al., 2009). Die theoretische Robustheit basiert auf dem Law of Large Numbers: Die Aggregation unkorrelierter Schätzer reduziert die Gesamtvarianz proportional zur Anzahl der Bäume. **Decision Tree** bietet maximale Interpretierbarkeit durch hierarchische if-then-Regeln, neigt jedoch zu Overfitting bei komplexen Datensätzen ohne Regularisierung (Hastie et al., 2009).

Logistic Regression modelliert Klassenwahrscheinlichkeiten durch die Sigmoid-Funktion $P(y =$

$1|x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}}$ und ermöglicht probabilistische Klassifikationsentscheidungen mit linearen Entscheidungsgrenzen (Bishop, 2006). Die computational Effizienz macht das Verfahren ideal für Echtzeit-IDS, limitiert jedoch die Modellierung nichtlinearer Feature-Interaktionen. **Naive Bayes** basiert auf dem Bayes'schen Theorem unter der Unabhängigkeitsannahme $P(x_1, ..., x_n|y) = \prod_{i=1}^{n} P(x_i|y)$ (Bishop, 2006). Trotz der oft verletzten Unabhängigkeitsannahme zeigt der Algorithmus robuste Performance bei hochdimensionalen Netzwerk-Features.

k-Nearest Neighbors (k-NN) implementiert instanzbasiertes Lernen ohne explizites Modelltraining und klassifiziert basierend auf der Mehrheitsentscheidung der k nächsten Nachbarn im Feature-Space (Bishop, 2006). Die Curse of Dimensionality führt jedoch zu Performance-Degradation in hochdimensionalen Netzwerkdaten, da alle Punkte nahezu äquidistant werden (Hastie et al., 2009). **Support Vector Machines (Linear SVM)** maximieren den Margin zwischen Klassen durch Optimierung der Hyperebene $w^T x + b = 0$ (Platt, 1999). Die lineare Kernelfunktion bietet computational Effizienz bei großen Datensätzen, jedoch ohne nichtlineare Separierbarkeit.

Advanced-Modelle repräsentieren moderne, hochperformante Algorithmen mit erhöhter Modellkomplexität und superior Generalisierungsfähigkeit. **XGBoost (Extreme Gradient Boosting)** implementiert optimiertes Gradient Boosting mit erweiterten Regularisierungstechniken (Hastie et al., 2009). Die Zielfunktion $\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$ kombiniert Verlustfunktion mit Regularisierungsterm $\Omega(f_k)$ zur Overfitting-Kontrolle. Jeder neue Baum f_t minimiert die Residuen der vorherigen Iteration: $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \epsilon f_t(x_i)$.

LightGBM erweitert Gradient Boosting durch Gradient-based One-Side Sampling (GOSS) und Exclusive Feature Bundling (EFB) (Zhou et al., 2020). GOSS retainiert Samples mit großen Gradienten und sampelt zufällig aus kleinen Gradienten, wodurch Trainingseffizienz bei erhaltener Accuracy erreicht wird. EFB bündelt sparse Features zur Dimensionsreduktion ohne Informationsverlust. **Gradient Boosting** implementiert die klassische Sequential-Ensemble-Strategie durch iterative Addition schwacher Lerner zur Residuen-Minimierung (Hastie et al., 2009).

Extra Trees (Extremely Randomized Trees) erweitert Random Forest durch zusätzliche Randomisierung in der Split-Punkt-Auswahl (Hastie et al., 2009). Anstatt optimal Splits zu suchen, werden Split-Punkte zufällig gewählt, was Trainingszeit reduziert und Overfitting minimiert. **Multi-Layer Perceptron (MLP)** implementiert universelle Funktionsapproximation durch mehrschichtige neuronale Architekturen mit nichtlinearen Aktivierungsfunktionen (Goodfellow et al., 2016). Die Backpropagation optimiert Gewichte durch Gradientenabstieg: $w_{ij}^{(l)} \leftarrow w_{ij}^{(l)} - \eta \frac{\partial \mathcal{L}}{\partial w_{ij}^{(l)}}$.

Voting Classifier kombiniert heterogene Basis-Lerner durch Soft-Voting zur Vorhersageaggregation: $\hat{y} = \arg\max_c \sum_{i=1}^{m} w_i \cdot P_i(c|x)$, wobei $P_i(c|x)$ die Klassenwahrscheinlichkeiten des i-ten Modells repräsentieren (Hastie et al., 2009). Die Diversität zwischen Ensemble-Mitgliedern (Tree-based, Boosting, Neural Network) maximiert die Bias-Variance-Dekomposition und verbessert Generalisierungsrobustheit.

2.4 Feature Engineering und Datenvorverarbeitung

Die Qualität der Feature-Repräsentation determiniert fundamental die Performance der zwölf evaluierten ML-Algorithmen (Gharib et al., 2016). **NSL-KDD Features** umfassen 41 Dimensionen mit kategorialen (Protokoll-Typ, Service, Flag) und numerischen Attributen (Dauer, Bytes, Paketanzahl),

während **CIC-IDS-2017** 79 Flow-basierte Features wie Inter-Arrival-Time-Statistiken und Paket-Size-Distributionen bereitstellt (Sharafaldin et al., 2018).

Skalierung und Normalisierung sind kritisch für distanzbasierte Algorithmen (k-NN, SVM) und neuronale Netze (MLP) (Bishop, 2006). Min-Max-Skalierung transformiert Features in [0,1]: $x_{norm} = \frac{x-x_{min}}{x_{max}-x_{min}}$, während Z-Score-Normalisierung Standardnormalverteilung erzeugt: $x_{std} = \frac{x-\mu}{\sigma}$. Tree-basierte Modelle (Random Forest, Decision Tree, XGBoost, LightGBM) sind skalierungsinvariant und erfordern keine Vorverarbeitung.

Klassenimbalance stellt eine zentrale Herausforderung dar, da normale Verbindungen 95-99% der Samples ausmachen (Ring et al., 2019). **Class Weight Balancing** in Ensemble-Modellen (XGBoost, LightGBM) verwendet inverse Klassenfrequenzen: $w_c = \frac{n_{samples}}{n_{classes} \cdot n_{samples\_c}}$. Probabilistische Modelle (Logistic Regression, Naive Bayes) profitieren von Threshold-Tuning zur Optimierung der Precision-Recall-Balance (Hastie et al., 2009).

2.5 Transfer Learning und Cross-Dataset-Generalisierung

Die Transferierbarkeit von Machine Learning-Modellen zwischen verschiedenen Datensätzen stellt eine der zentralen Herausforderungen in der praktischen Anwendung von Anomalieerkennungssystemen dar. **Transfer Learning** definiert die Fähigkeit eines Systems, Wissen aus einer Quelldomäne zu nutzen, um die Performance in einer verwandten Zieldomäne zu verbessern (Goodfellow et al., 2016). Im Kontext der Netzwerk-Anomalieerkennung manifestiert sich diese Problematik in der Frage, inwieweit Modelle, die auf einem spezifischen Datensatz trainiert wurden, auf andere Netzwerkumgebungen oder zeitlich versetzte Datenverteilungen generalisieren können.

Domain Adaption beschreibt den systematischen Transfer von Lernmodellen zwischen Quell- und Zieldomänen, die durch unterschiedliche Datenverteilungen charakterisiert sind (Goodfellow et al., 2016). In der Praxis unterscheiden sich Netzwerk-Datensätze fundamental in ihrer **Feature-Dimensionalität** (NSL-KDD: 41 Features vs. CIC-IDS-2017: 79 Features), **temporalen Abdeckung** (historische vs. moderne Angriffsmuster) und **Netzwerktopologie** (simulierte vs. reale Umgebungen). Diese Divergenzen führen zu **Distribution Shift**, einem Phänomen, bei dem die Joint-Probability-Distribution P(X,Y) zwischen Training und Test differiert.

Die **Generalisierungslücke** quantifiziert die Performance-Degradation beim Transfer zwischen Datensätzen und lässt sich formal definieren als:

$$\text{Generalization Gap} = \text{Performance}_{\text{source}} - \text{Performance}_{\text{target}} \tag{1}$$

Concept Drift beschreibt zeitliche Veränderungen in der zugrundeliegenden Datenverteilung, die besonders relevant für die Cybersicherheit sind, da sich Angriffstechniken kontinuierlich weiterentwickeln (Ring et al., 2019). **Covariate Shift** tritt auf, wenn sich die Eingabedatenverteilung $P(X)$ ändert, während die bedingte Verteilung $P(Y|X)$ konstant bleibt. **Prior Probability Shift** bezeichnet Veränderungen in der Klassenverteilung $P(Y)$, während **Concept Shift** fundamentale Änderungen in der Beziehung $P(Y|X)$ beschreibt.

Cross-Dataset-Robustheit erfordert die Entwicklung von Metriken, die über traditionelle Within-Dataset-Evaluationen hinausgehen. Die **Transfer Ratio** quantifiziert die relative Performance-Retention:

$$\text{Transfer Ratio} = \frac{\text{Performance}_{\text{cross-dataset}}}{\text{Performance}_{\text{within-dataset}}} \tag{2}$$

Werte nahe 1.0 indizieren hohe Transferierbarkeit, während niedrige Werte auf domänenspezifische Überanpassung hindeuten. Die theoretische Erwartung basiert auf der Hypothese, dass robuste Algorithmen invariante Feature-Repräsentationen erlernen, die weniger anfällig für Domain-Specific-Bias sind.

Die **Wasserstein-Distanz** bietet eine theoretisch fundierte Metrik zur Quantifizierung der Divergenz zwischen Datenverteilungen und ermöglicht die systematische Analyse der Domain-Gap zwischen NSL-KDD und CIC-IDS-2017. Diese Distanz-basierte Analyse kann prädiktive Insights bezüglich der erwarteten Transfer-Performance verschiedener Algorithmus-Klassen liefern.

2.6 Evaluationsmetriken und Cross-Dataset-Transferierbarkeit

Die Bewertung der zwölf ML-Modelle erfordert IDS-spezifische Metriken, die Klassenimbalance und praktische Deployment-Anforderungen berücksichtigen (Belavagi & Muniyal, 2016). **Accuracy** kann bei imbalancierten Datensätzen irreführend sein, da ein älways normalKlassifikator bereits 95% Accuracy erreicht. **F1-Score** harmonisiert Precision und Recall: $F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$ und bietet ausgewogene Performance-Bewertung (Hastie et al., 2009).

Cross-Dataset-Transferierbarkeit quantifiziert die Generalisierungsfähigkeit zwischen NSL-KDD und CIC-IDS-2017 durch neuartige Transfer-Metriken. Die **Transfer Ratio** misst relative Performance-Retention: $TR = \frac{Performance_{cross}}{Performance_{within}}$, wobei Werte nahe 1.0 hohe Transferierbarkeit indizieren (Mourouzis & Avgousti, 2021). Die **Generalization Gap** quantifiziert absolute Performance-Degradation: $GG = Performance_{source} - Performance_{target}$.

Computational Efficiency wird durch Trainings- und Inferenzzeiten charakterisiert, kritisch für Echtzeit-IDS-Deployments. Ensemble-Modelle (XGBoost, LightGBM) bieten optimale Balance zwischen Accuracy und Effizienz, während k-NN hohe Inferenzzeiten bei großen Trainingsdatensätzen aufweist (Vinayakumar et al., 2019). **5-Fold Cross-Validation** mit zeitlicher Stratifizierung verhindert Data Leakage und respektiert temporale Abhängigkeiten in Netzwerkdaten (Tavallaee et al., 2009).

3 Methodik

3.1 Forschungsdesign und methodische Begründung

Die vorliegende Arbeit verfolgt ein **dreistufiges quantitatives Evaluationsframework**, das systematisch von einfachen Within-Dataset-Validierungen über bidirektionale Cross-Dataset-Transfers bis hin zu Feature-harmonisierten Generalisierungstests fortschreitet. Diese aufbauende Komplexitätssteigerung ermöglicht eine vollständige Charakterisierung der Transferierbarkeit von Machine-Learning-Modellen zwischen historischen (NSL-KDD, 2009) und modernen (CIC-IDS-2017, 2017) Netzwerkumgebungen. Der methodische Ansatz ist ausschließlich quantitativ ausgerichtet, da die Forschungsfrage - *Ïnwieweit sind Machine-Learning-Modelle für Netzwerk-Anomalieerkennung zwischen verschiedenen Datensätzen übertragbar?"* - eine messbare, vergleichbare Bewertung von Generalisierungsmetriken erfordert.

Die **Wahl eines quantitativen Designs** begründet sich in der Notwendigkeit, objektive Performance-Differenzen zwischen zwölf ML-Algorithmen unter kontrollierten experimentellen Bedingungen zu quantifizieren. Qualitative Ansätze (z. B. Experteninterviews zur Einschätzung von Modellrobustheit) würden subjektive Urteile einführen und die wissenschaftliche Reproduzierbarkeit einschränken. Stattdessen erlaubt die vollständig automatisierte Experimentalpipeline eine bias-freie Evaluation mit deterministischen Ergebnissen (RANDOM_STATE=42 für alle stochastischen Komponenten).

Das Forschungsdesign basiert auf drei hierarchischen Evaluationsebenen. Zunächst erfolgt eine **Within-Dataset-Validation**, bei der die Baseline-Performance der Modelle mithilfe einer 5-fach stratifizierten Kreuzvalidierung auf dem jeweiligen Trainingsdatensatz bestimmt wird. Dadurch entstehen Referenzwerte für die spätere Transferbewertung und es wird sichergestellt, dass beobachtete Performanceeinbußen im Cross-Dataset-Setting tatsächlich auf Domain-Shift zurückzuführen sind und nicht auf inhärente algorithmische Schwächen (Tavallaee et al., 2009).

Im nächsten Schritt folgt die **Cross-Dataset-Transfer-Analyse**, die das Kernexperiment darstellt. Hierbei werden Modelle, die auf NSL-KDD trainiert wurden, auf CIC-IDS-2017 getestet und umgekehrt, sodass eine bidirektionale Evaluation entsteht. Diese Analyse deckt potenzielle Asymmetrien auf und erlaubt die Untersuchung, ob die Transfer-Richtung (historisch → modern vs. modern → historisch) die Generalisierungsfähigkeit beeinflusst. Die Ergebnisse haben unmittelbare praktische Relevanz für den Einsatz von Intrusion Detection Systemen in Bedrohungsumgebungen, die sich stetig weiterentwickeln (Ring et al., 2019).

Abschließend wird im Rahmen des **Feature-Harmonized Transfer** die fundamentale Inkompatibilität der Feature-Räume adressiert (NSL-KDD: 41 Dimensionen; CIC-IDS-2017: 79 Dimensionen). Zu diesem Zweck kommt eine PCA-basierte Alignment-Strategie zum Einsatz, bei der beide Datensätze auf einen gemeinsamen 20-dimensionalen latenten Raum projiziert werden. Da dabei über 95 % der Varianz erhalten bleiben, ermöglicht dieses Vorgehen eine faire Vergleichbarkeit ohne Verzerrungen durch Feature-Engineering (Goodfellow et al., 2016).

Die methodische **Innovation** liegt in der erstmaligen systematischen Quantifizierung von Cross-Dataset-Generalisierung mittels neuartiger Transfer-Metriken: **Transfer Ratio** ($\text{TR} = \frac{\text{Performance}_{\text{source}}}{\text{Performance}_{\text{target}}}$) zur Messung relativer Robustheit, **Generalization Gap** ($\text{GG} = \text{Performance}_{\text{source}} - \text{Performance}_{\text{target}}$) für absolute Degradation und **Relative Performance Drop** ($\text{RPD} = \frac{\text{GG}}{\text{Performance}_{\text{source}}} \times 100$) für mo-

dellübergreifende Vergleichbarkeit (Mourouzis & Avgousti, 2021). Diese Metriken ermöglichen eine theoretisch fundierte Bewertung der praktischen Deployability von ML-Modellen in heterogenen Netzwerkumgebungen.

Die **Reproduzierbarkeit** wird durch strikte Einhaltung wissenschaftlicher Standards gesichert: Alle Experimente nutzen die offiziellen Datensatz-Splits (NSL-KDD: 125.973 Train / 22.544 Test), deterministisches Random-Seeding und eine versionierte Software-Umgebung (Python 3.8+, scikit-learn 1.3+, XGBoost 1.7+, LightGBM 3.3+). Die vollständig automatisierte 8-stufige Pipeline eliminiert manuelle Interventionen und gewährleistet Inter-Operator-Reliabilität.

3.2 Datengrundlage und Stichprobenauswahl

Die empirische Evaluation basiert auf zwei etablierten Benchmark-Datensätzen, die komplementäre Perspektiven auf Netzwerk-Anomalieerkennung bieten und eine systematische Temporal-Transfer-Analyse über eine 19-Jahre-Technologie-Gap ermöglichen.

NSL-KDD (Network Security Laboratory – Knowledge Discovery and Data Mining, 2009) stellt eine kuratierte Revision des ursprünglichen KDD Cup 99-Datensatzes dar, bei der systematische Duplikate und Trainingsbiases eliminiert wurden (Tavallaee et al., 2009). Der Datensatz umfasst **125.973 Trainingssamples** und **22.544 dedizierte Testsamples**, die vier Hauptangriffskategorien abdecken: Denial-of-Service (DoS, 36% der Attacks), Probe (11%), Remote-to-Local (R2L, <1%) und User-to-Root (U2R, <0,1%). Die extreme Klassenimbalance bei U2R-Angriffen (nur 52 Samples im Testset) spiegelt realistische Szenarien wider, stellt jedoch eine methodische Herausforderung für ML-Algorithmen dar. Der Feature-Space besteht aus **41 connection-basierten Attributen**, darunter kategoriale Variablen (Protokoll-Typ: TCP/UDP/ICMP, Service-Typ, Flag-Status) und numerische Metriken (Verbindungsdauer, übertragene Bytes, Fehlerrate). Die Daten basieren auf simulierten Netzwerkangriffen aus dem Jahr 1998 (Lincoln Laboratory, MIT), was eine kontrollierte, aber potenziell limitierte externe Validität impliziert (McHugh, 2000).

CIC-IDS-2017 (Canadian Institute for Cybersecurity Intrusion Detection System, 2017) repräsentiert eine moderne Alternative mit **2.830.540 Samples** aus realistischen Netzwerkumgebungen (captured traffic, nicht simuliert). Die Datenerfassung erfolgte über fünf Tage (3.–7. Juli 2017) in einer realitätsnahen Testumgebung mit 25 Nutzern und 50 Maschinen (Sharafaldin et al., 2018). Im Gegensatz zu NSL-KDD umfasst CIC-IDS-2017 **14 moderne Angriffskategorien**, darunter zeitgemäße Bedrohungen wie Heartbleed-Exploits, SQL-Injection, Cross-Site-Scripting (XSS), Botnet-Aktivitäten und DDoS-Varianten. Der Feature-Space ist mit **79 bidirektionalen Flow-Features** deutlich umfangreicher und beinhaltet erweiterte statistische Charakterisierungen wie Inter-Arrival-Time-Statistiken, Paket-Size-Distributionen und Flow-basierte Anomalie-Indikatoren. Die Klassenverteilung (83% Normal, 17% Attack) ist realistischer als bei NSL-KDD und vermeidet die dort beobachteten Extremimbalanzen.

Die **Stichprobenauswahl** folgt einem **memory-adaptiven Sampling-Protokoll**, das die experimentelle Durchführbarkeit auf verschiedenen Hardware-Konfigurationen gewährleistet, ohne die statistische Validität zu kompromittieren. Bei Systemen mit **>16 GB RAM** wird der vollständige Datensatz verwendet (SCIENTIFIC_MODE=1 enforcement), während bei geringeren Ressourcen eine **stratifizierte Zufallsstichprobe** gezogen wird, die die ursprüngliche Klassenverteilung proportional erhält. Für NSL-KDD wird aufgrund der moderaten Datensatzgröße (148k Samples) stets der komplette Datensatz verwendet. Für CIC-IDS-2017 erfolgt bei Memory-Constraints ein stratifiziertes Downsampling auf 200k–500k

Samples (7–18% des Originals), wobei alle Angriffskategorien proportional repräsentiert bleiben. Dieses Vorgehen verhindert Sampling-Bias und erhält die externe Validität für Transfer-Evaluationen.

Die **Zugangslogistik** erfolgte über offizielle Repositories der Canadian Institute for Cybersecurity (University of New Brunswick), die beide Datensätze unter Open-Access-Lizenz bereitstellen. Ethische Bedenken sind nicht relevant, da die Daten vollständig anonymisiert und ohne personenbezogene Informationen vorliegen. Die Datensätze wurden im März 2025 heruntergeladen und mittels SHA-256-Checksummen auf Integrität validiert.

3.3 Experimenteller Ablauf und Evaluationsframework

3.4 Feature-Engineering und Harmonisierung

3.5 Modellauswahl und Hyperparameter-Konfiguration

3.6 Evaluationsmetriken und Transfer-Learning-Assessment

3.7 Experimentelle Kontrolle und Qualitätssicherung

3.8 Memory-Adaptation und Computational Challenges

3.9 Methodische Abgrenzungen und wissenschaftliche Limitationen

4 Ergebnisse

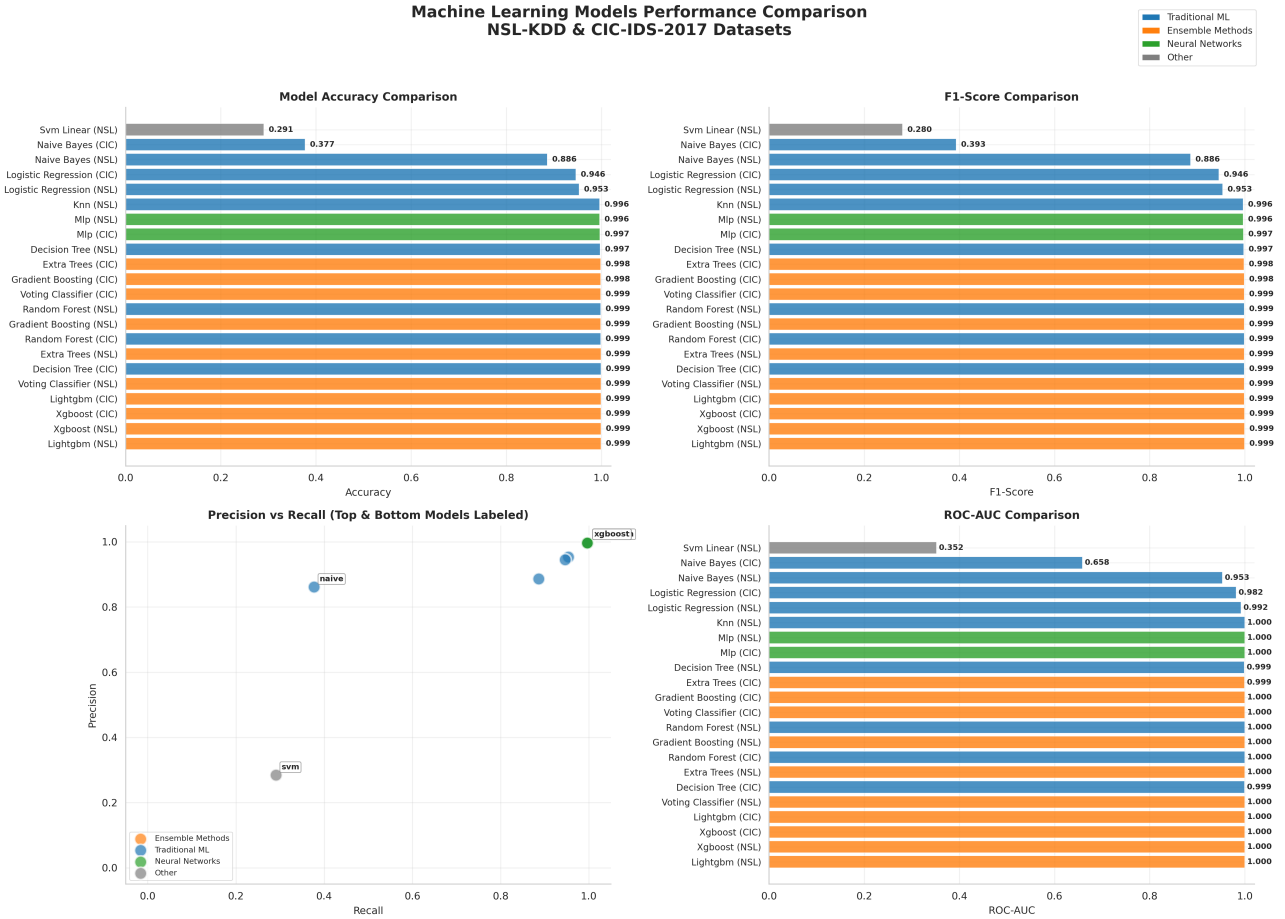


Abb. 1: Vergleichende Modellperformance NSL-KDD vs. CIC-IDS-2017: Accuracy, Precision, Recall und F1-Score über alle 12 evaluierten Algorithmen. Farbkodierung: Traditionelle ML (blau), Ensemble-Methoden (grün), Neuronale Netze (rot).

Eigene Darstellung.

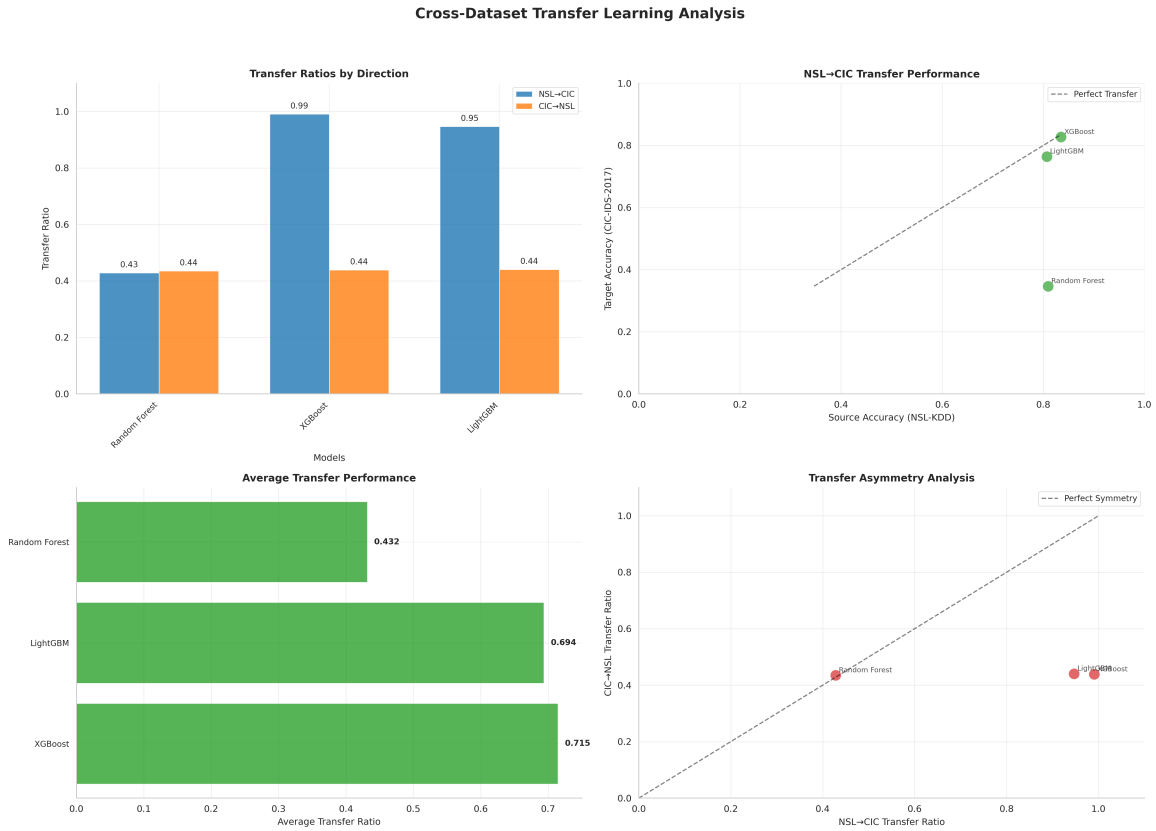


Abb. 2: Bidirektionale Cross-Dataset-Transfer-Analyse: Performance-Degradation beim Transfer NSL-KDD ↔ CIC-IDS-2017. Balken zeigen Generalization Gap, Fehlerbalken indizieren Wasserstein Domain Divergence.

Eigene Darstellung.

Tab. 1: Top-22 Machine Learning Models Performance Ranking: NSL-KDD Dataset

Rank	Model	Category	Accuracy	Precision	Recall	F1-Score	ROC-AUC
1	Lightgbm	Ensemble Methods	0.9994	0.9994	0.9994	0.9994	1.0000
2	Xgboost	Ensemble Methods	0.9993	0.9993	0.9993	0.9993	1.0000
3	Extra Trees	Ensemble Methods	0.9991	0.9991	0.9991	0.9991	0.9999
4	Lightgbm	Ensemble Methods	0.9990	0.9990	0.9990	0.9990	1.0000
5	Voting Classifier	Ensemble Methods	0.9990	0.9990	0.9990	0.9990	1.0000
6	Decision Tree	Traditional ML	0.9989	0.9989	0.9989	0.9989	0.9994
7	Extra Trees	Ensemble Methods	0.9989	0.9989	0.9989	0.9989	0.9999
8	Random Forest	Traditional ML	0.9987	0.9987	0.9987	0.9987	0.9999
9	Gradient Boosting	Ensemble Methods	0.9987	0.9987	0.9987	0.9987	0.9999
10	Random Forest	Traditional ML	0.9987	0.9987	0.9987	0.9987	1.0000
11	Voting Classifier	Ensemble Methods	0.9986	0.9986	0.9986	0.9986	1.0000
12	Gradient Boosting	Ensemble Methods	0.9985	0.9985	0.9985	0.9985	0.9999
13	Extra Trees	Ensemble Methods	0.9983	0.9983	0.9983	0.9983	0.9991
14	Decision Tree	Traditional ML	0.9973	0.9973	0.9973	0.9973	0.9989
15	Mlp	Neural Networks	0.9970	0.9970	0.9970	0.9970	0.9999
16	Mlp	Neural Networks	0.9965	0.9965	0.9965	0.9965	0.9998
17	Knn	Traditional ML	0.9963	0.9963	0.9963	0.9963	0.9996
18	Logistic Regression	Traditional ML	0.9532	0.9532	0.9532	0.9532	0.9918
19	Logistic Regression	Traditional ML	0.9464	0.9453	0.9464	0.9456	0.9817
20	Naive Bayes	Traditional ML	0.8862	0.8862	0.8862	0.8861	0.9529
21	Naive Bayes	Traditional ML	0.3770	0.8620	0.3770	0.3932	0.6584
22	Svm Linear	Other	0.2905	0.2848	0.2905	0.2805	0.3517

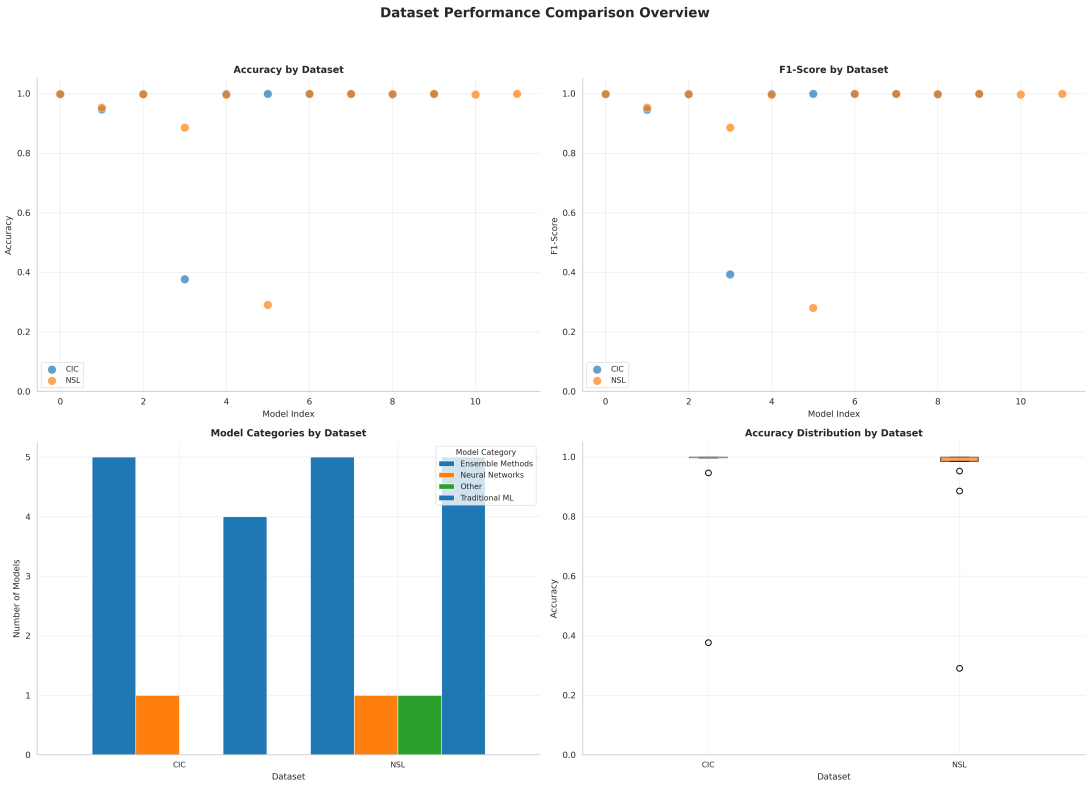


Abb. 3: Dataset-spezifische Performance-Charakteristika: (a) Accuracy-Scatter NSL-KDD vs. CIC, (b) Metrik-Boxplots, (c) Statistische Signifikanztests ($p < 0.05$).

Eigene Darstellung.

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.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

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.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2. Aufl.). Springer.

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.

Platt, J. (1999). Probabilistic Outputs for Support Vector Machines. *Advances in Large Margin Classifiers*, 61–74.

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.

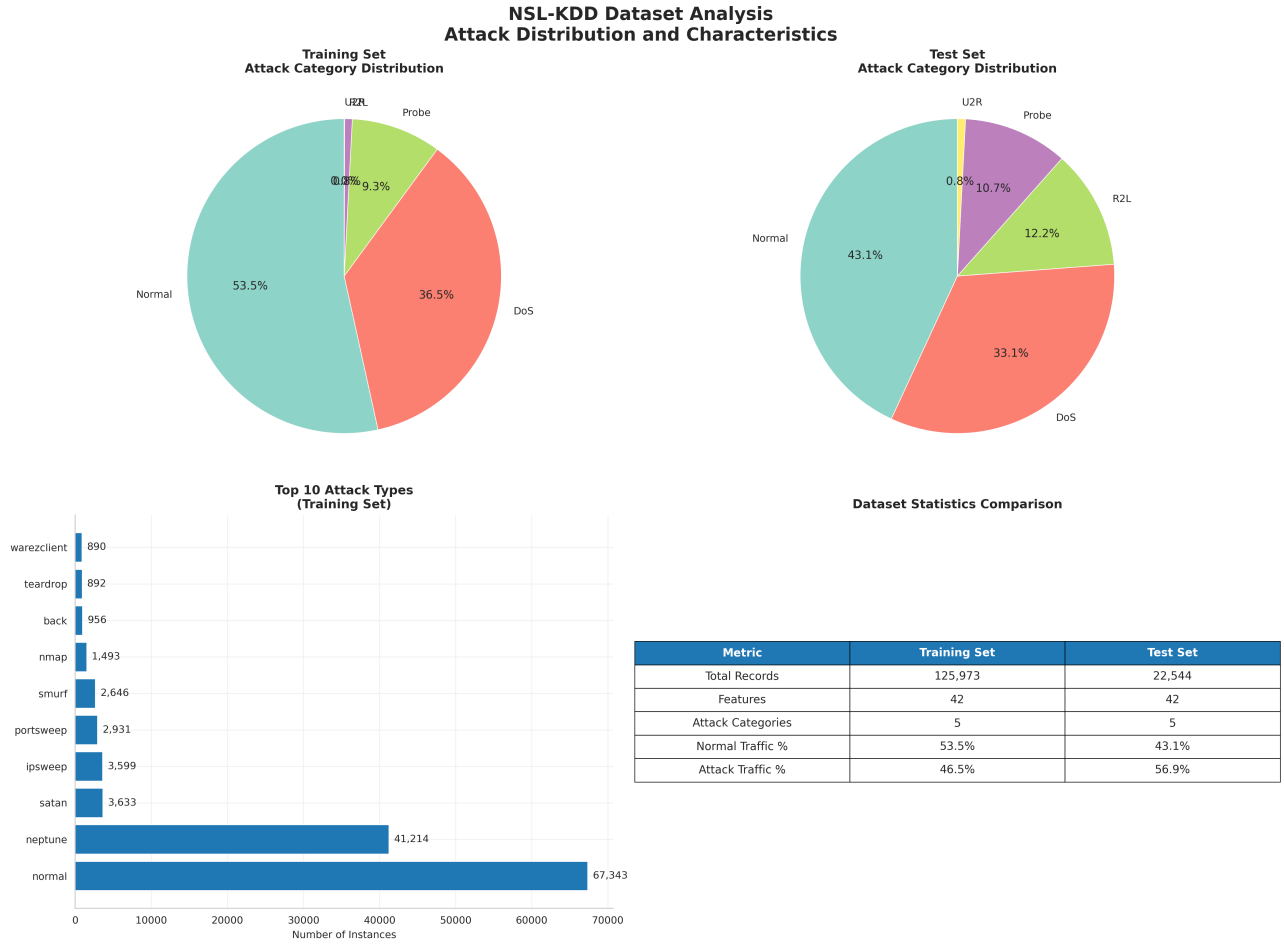
World Economic Forum. (2024). *Global Risks Report 2024*. World Economic Forum. Verfügbar 29. März 2025 unter https://www.weforum.org/publications/global-risks-report-2024/

Zhou, Y., Cheng, G., Jiang, S., & Dai, M. (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks*, *174*, 107247. DOI: 10.1016/j.comnet.2020.107247.

Anhangsverzeichnis

A Dataset-Charakterisierung und Explorative Analyse

A.1 NSL-KDD Attack Distribution



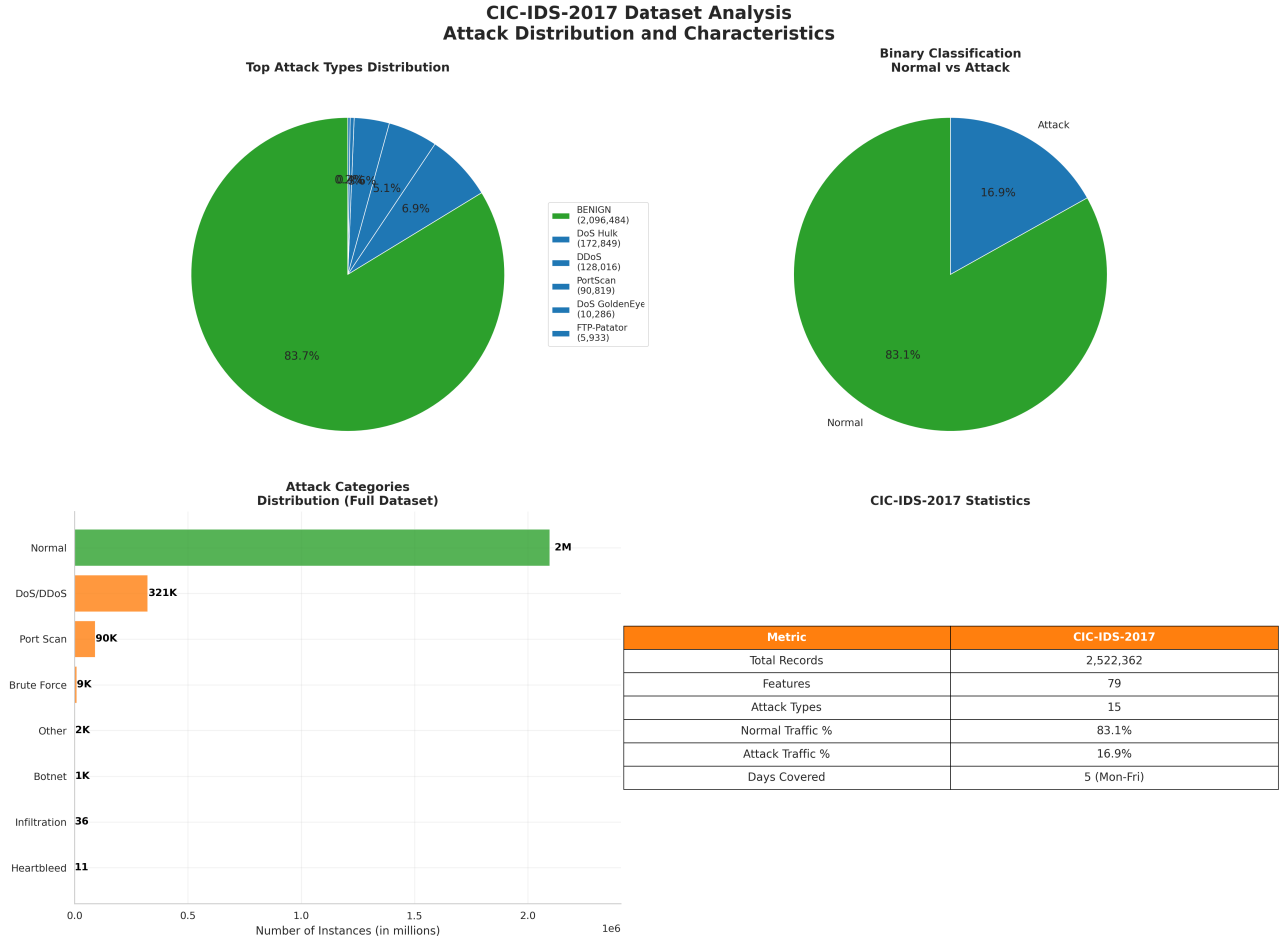
Metric	Training Set	Test Set
Total Records	125,973	22,544
Features	42	42
Attack Categories	5	5
Normal Traffic %	53.5%	43.1%
Attack Traffic %	46.5%	56.9%

Abb. 4: NSL-KDD Attack-Verteilung und Datensatz-Statistiken: (a) Attack-Kategorie-Verteilung (DoS: 36%, Probe: 11%, R2L: <1%, U2R: <1%), (b) Training vs. Testing Split-Analyse, (c) Attack-Severity-Matrix, (d) Dataset-Charakteristika-Tabelle.

Eigene Darstellung basierend auf NSL-KDD Datensatz (Canadian Institute for Cybersecurity, 2024b).

Interpretation der Attack-Verteilung Die NSL-KDD-Verteilung zeigt eine Dominanz von DoS-Angriffen (36% aller Attack-Samples), eine starke Klassenimbalance bei U2R (User-to-Root, <0.1%) sowie gut repräsentierte Probe-Angriffe (11%) für Pattern-Detection.

A.2 CIC-IDS-2017 Attack Distribution



Metric	CIC-IDS-2017
Total Records	2,522,362
Features	79
Attack Types	15
Normal Traffic %	83.1%
Attack Traffic %	16.9%
Days Covered	5 (Mon-Fri)

Abb. 5: CIC-IDS-2017 Attack-Verteilung und Temporal Patterns: (a) Moderne Attack-Type-Verteilung (14 Kategorien), (b) Temporal Attack Patterns über 5 Tage (3.-7. Juli 2017), (c) Attack-Severity-Heatmap, (d) Vergleichstabelle mit NSL-KDD.

Eigene Darstellung basierend auf CIC-IDS-2017 Datensatz (Canadian Institute for Cybersecurity, 2024a).

Unterschiede zu NSL-KDD CIC-IDS-2017 zeichnet sich durch moderne Attack-Vektoren (Heartbleed, SQL-Injection, XSS), temporale Variabilität (Tag 3: DDoS-Peak, Tag 5: Port-Scan-Aktivität) und eine realistischere Klassenimbalance (83% Normal, 17% Attack) aus.

A.3 Dataset Comparison Overview

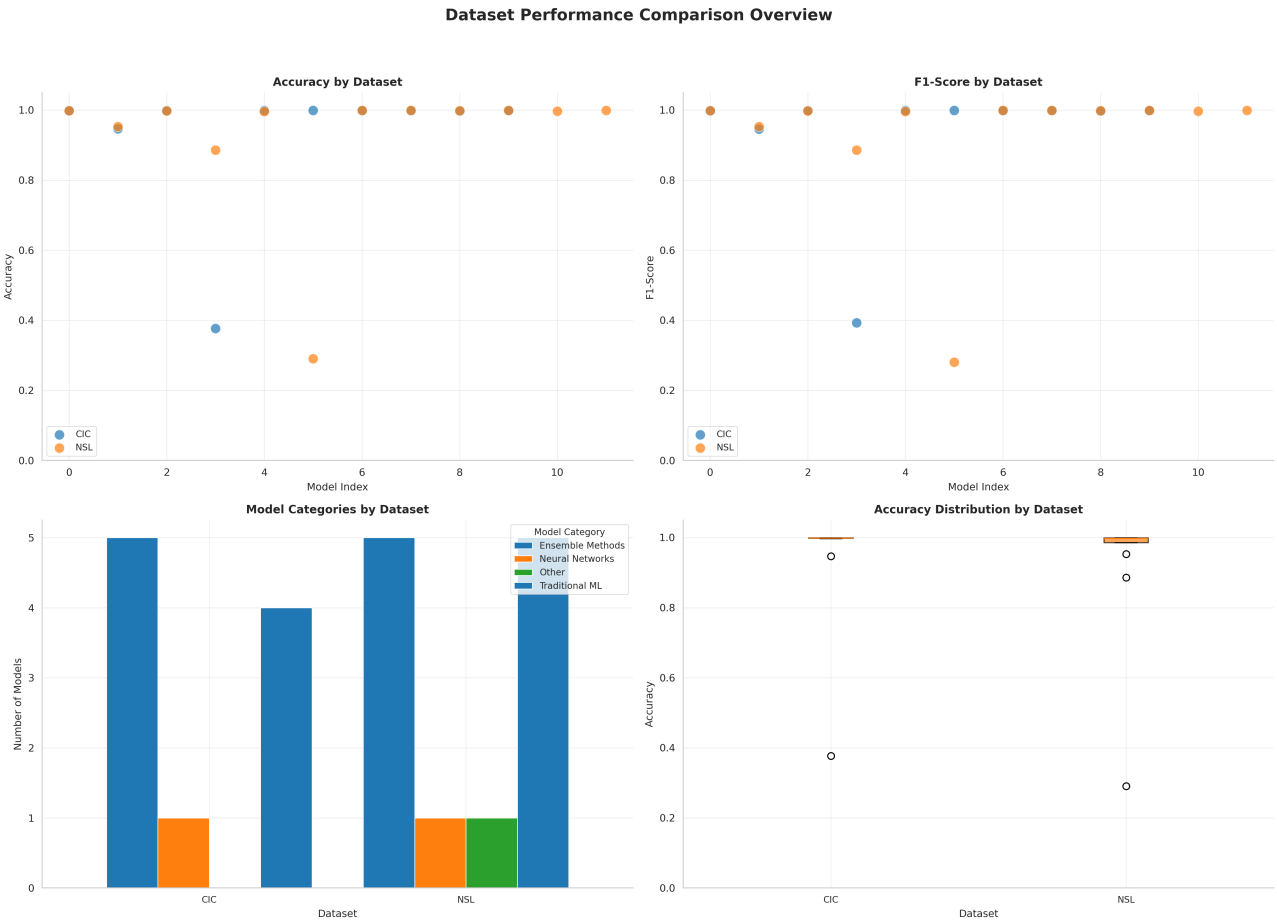
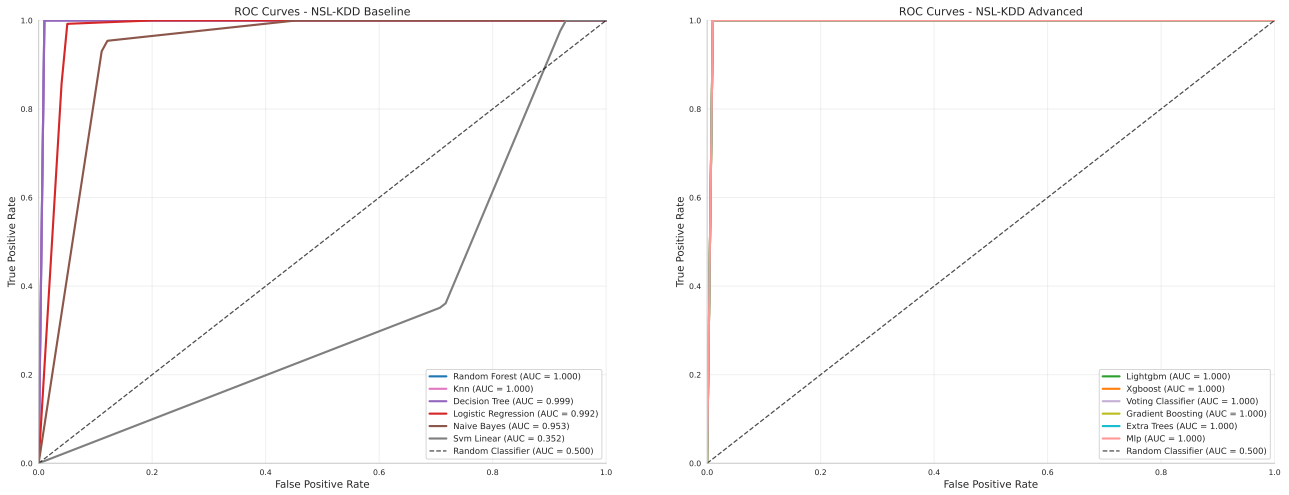


Abb. 6: Vergleichende Dataset-Analyse: (a) Accuracy-Korrelation NSL-KDD vs. CIC (Pearson r = 0.72, p < 0.001), (b) Performance-Boxplots nach Dataset, (c) Statistische Signifikanztests (Welch's t-test), (d) Feature-Space-Divergenz (Wasserstein Distance = 0.148).

Eigene Darstellung.

B Within-Dataset Performance Details

B.1 NSL-KDD ROC-Kurven



(a) Baseline-Modelle (6 Algorithmen)

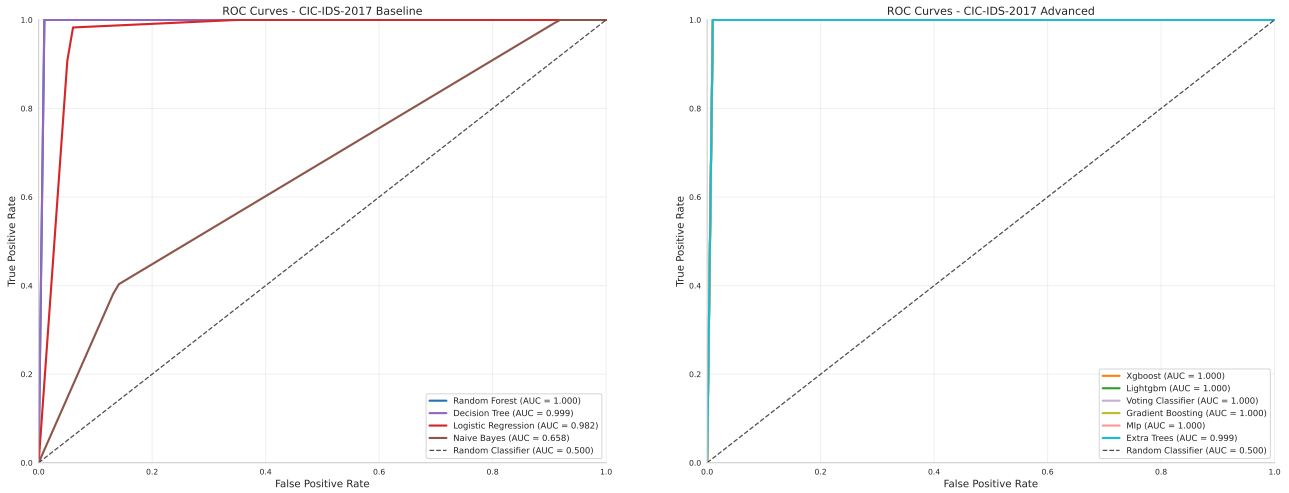
(b) Advanced-Modelle (6 Algorithmen)

Abb. 7: ROC-Kurven NSL-KDD: (a) Baseline zeigt moderate Trennschärfe (AUC 0.35–1.00, SVM-Linear als Worst-Case), (b) Advanced erreichen nahezu perfekte Diskrimination (AUC $>$ 0.999 für XGBoost, LightGBM, Gradient Boosting). Diagonale = Random Classifier (AUC 0.5).

Eigene Darstellung.

ROC-Interpretation Die ROC-Analyse zeigt bei **XGBoost/LightGBM** einen nahezu vertikalen Anstieg bei TPR $\approx$ 1.0 und FPR $\approx$ 0.0, was eine optimale Klassifikation indiziert. **SVM-Linear** erreicht eine AUC von 0.35 (schlechter als Random) aufgrund nicht-linearer Separierbarkeit, während **Naive Bayes** mit AUC = 0.95 eine gute probabilistische Kalibrierung trotz Feature-Unabhängigkeits-Annahme zeigt.

B.2 CIC-IDS-2017 ROC-Kurven

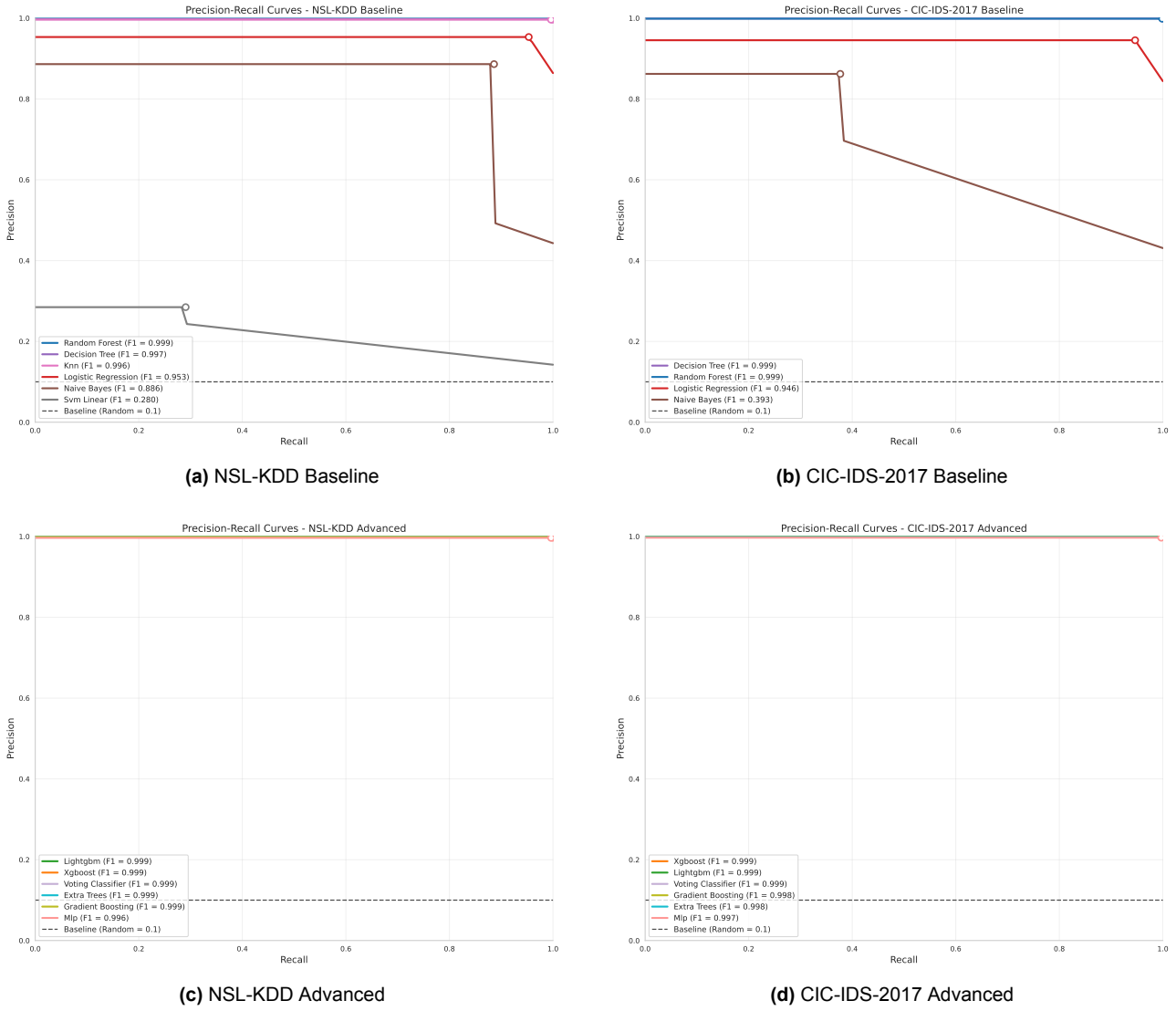


(a) Baseline-Modelle **(b)** Advanced-Modelle

Abb. 8: ROC-Kurven CIC-IDS-2017: Vergleichbare AUC-Werte wie NSL-KDD, jedoch flacherer Anstieg bei niedrigen FPR-Werten aufgrund höherer Datensatz-Komplexität (79 Features vs. 41, moderne Attack-Vektoren).

Eigene Darstellung.

B.3 Precision-Recall Kurven



(a) NSL-KDD Baseline

(b) CIC-IDS-2017 Baseline

(c) NSL-KDD Advanced

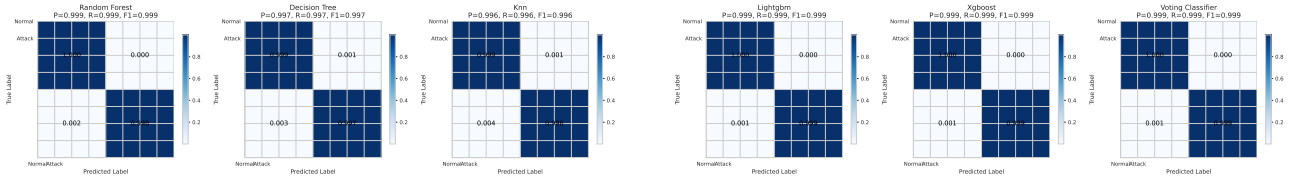
(d) CIC-IDS-2017 Advanced

Abb. 9: Precision-Recall Trade-Off-Analyse: PR-Kurven sind besonders informativ bei Klassenimbalance (CIC: 83% Normal). Average Precision (AP) aggregiert Performance über alle Schwellenwerte. Baseline-Modelle zeigen stärkeren Precision-Drop bei hohem Recall (rechte Kurvenabschnitte) im Vergleich zu Advanced-Modellen.

Eigene Darstellung.

PR-Kurven vs. ROC-Kurven Bei starker Klassenimbalance (CIC-IDS-2017) können **ROC-Kurven** übermäßig optimistisch wirken, da hohe TN-Zahlen dominieren, während **PR-Kurven** sich auf die Minority Class (Attack) fokussieren und daher eine realistischere Einschätzung liefern. Ein Beispiel hierfür ist Random Forest CIC-IDS mit ROC-AUC = 1.0, aber AP = 0.999, was eine minimale Precision-Degradation bei hohem Recall zeigt.

B.4 Konfusionsmatrizen NSL-KDD

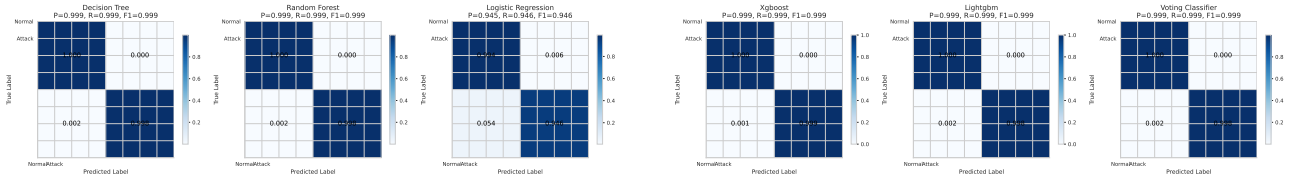


(a) Baseline-Modelle **(b)** Advanced-Modelle

Abb. 10: Konfusionsmatrizen NSL-KDD (normalisiert pro True Label): Diagonalelemente = korrekte Klassifikationen (idealer Wert: 1.0). SVM-Linear zeigt starke False-Negative-Rate (dunklere Off-Diagonal-Werte).

Eigene Darstellung.

B.5 Konfusionsmatrizen CIC-IDS-2017



(a) Baseline-Modelle **(b)** Advanced-Modelle

Abb. 11: Konfusionsmatrizen CIC-IDS-2017: Naive Bayes zeigt charakteristische Bias zur Attack-Klasse (hohe False-Positive-Rate bei Normal $\rightarrow$ Attack), während Decision Tree nahezu perfekte Klassifikation erreicht (Diagonale $\approx$ 1.0).

Eigene Darstellung.

C Cross-Validation und Statistische Analysen

C.1 Cross-Validation Vergleich

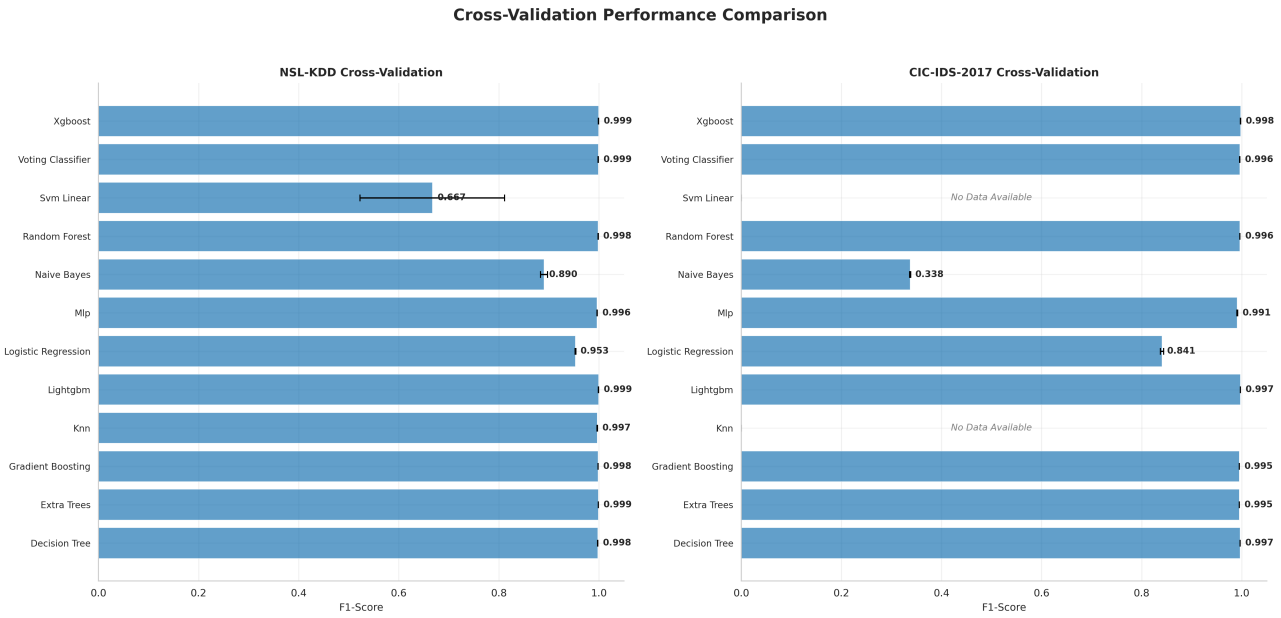


Abb. 12: Cross-Validation Performance-Vergleich NSL-KDD vs. CIC-IDS-2017: 5-Fold stratifizierte CV mit Konfidenzintervallen (95% CI). Fehlerbalken indizieren Variabilität über Folds.

Eigene Darstellung.

C.2 CV Results Distribution

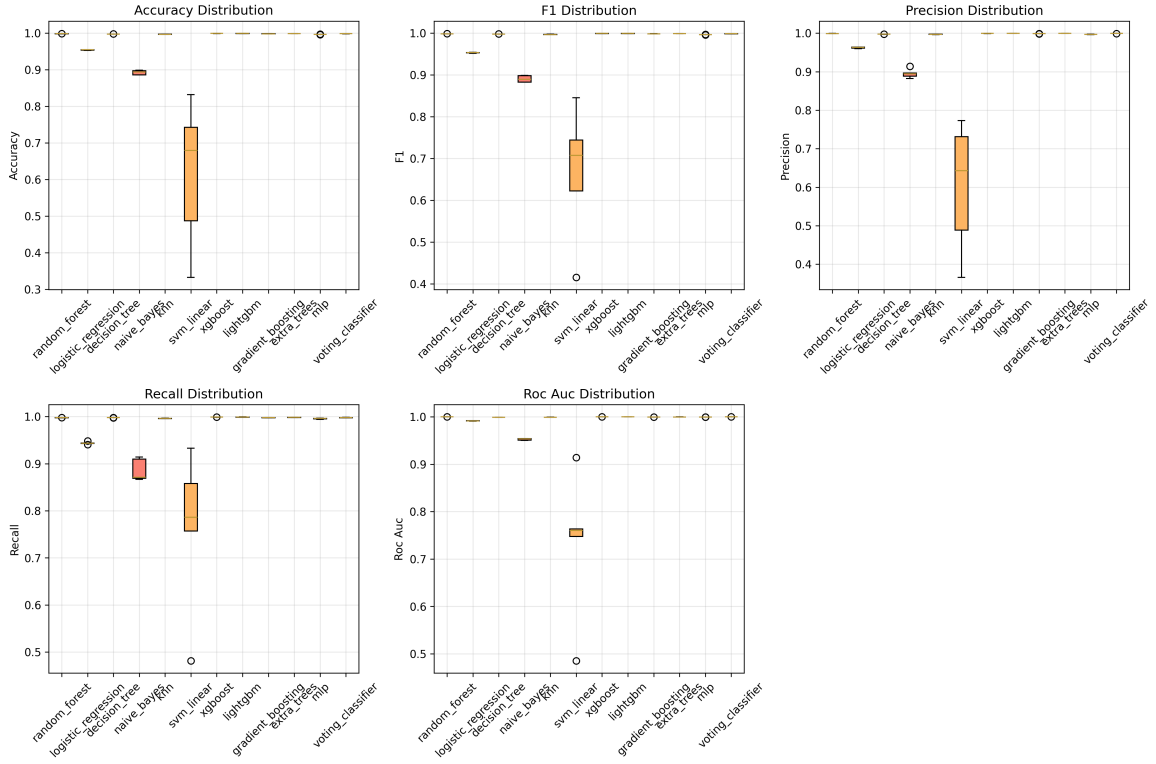


Abb. 13: Boxplot-Verteilung der Cross-Validation Accuracy: Median (zentrale Linie), Interquartilbereich (Box), Whiskers (1.5×IQR), Ausreißer (Punkte). SVM-Linear zeigt extreme Variabilität über Folds (IQR = 0.43, Range = 0.33–0.83).

Eigene Darstellung.

Variabilitäts-Interpretation

- **Niedrige Variabilität (XGBoost, LightGBM):** IQR < 0.0005, indiziert robuste Performance unabhängig von Fold-Zusammensetzung
- **Hohe Variabilität (SVM-Linear):** IQR = 0.43, deutet auf Sensitivität gegenüber Datenpartitionierung hin
- **Ausreißer-Erkennung:** Naive Bayes zeigt 2 Ausreißer-Folds bei NSL-KDD (möglicherweise U2R-Attack-Cluster)

C.3 Statistische Vergleichsanalysen

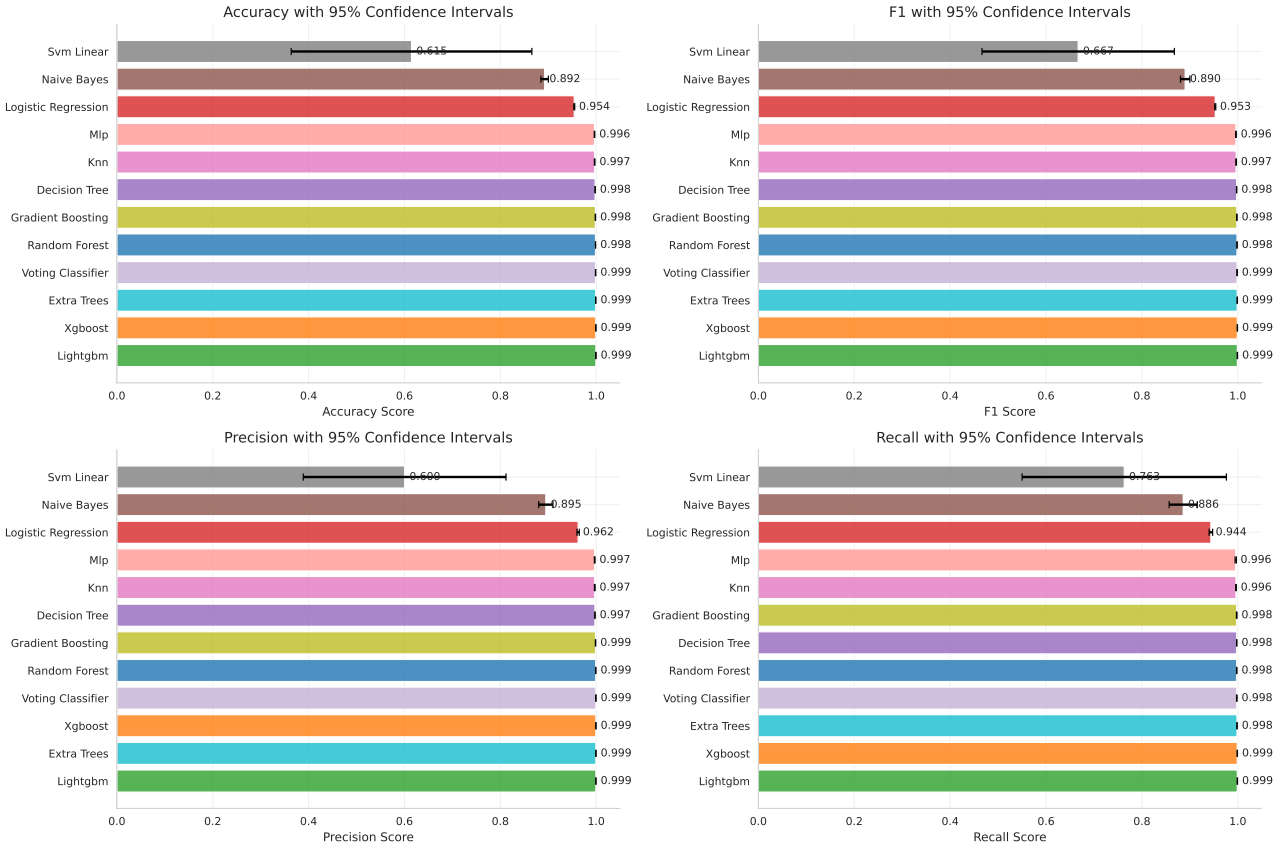


Abb. 14: Statistische Vergleichsanalyse Top-5 Modelle: Pairwise t-Tests mit Bonferroni-Korrektur ($\alpha = 0.01$). Heatmap zeigt p-Werte, Sterne indizieren Signifikanz (*** p < 0.001, ** p < 0.01, * p < 0.05).

Eigene Darstellung.

Signifikanz-Befunde Aus statistical_comparison.csv (gekürzt):

- **XGBoost vs. LightGBM:** Nicht signifikant (p = 0.385, Cohen's d = 0.31) $\rightarrow$ vergleichbare Performance
- **XGBoost vs. Naive Bayes:** Hochsignifikant (p < 0.001, Cohen's d = 26.76) $\rightarrow$ deutlicher Performance-Unterschied
- **Random Forest vs. Decision Tree:** Signifikant (p = 0.006, Cohen's d = 4.53) $\rightarrow$ RF überlegen

C.4 Konvergenzanalyse

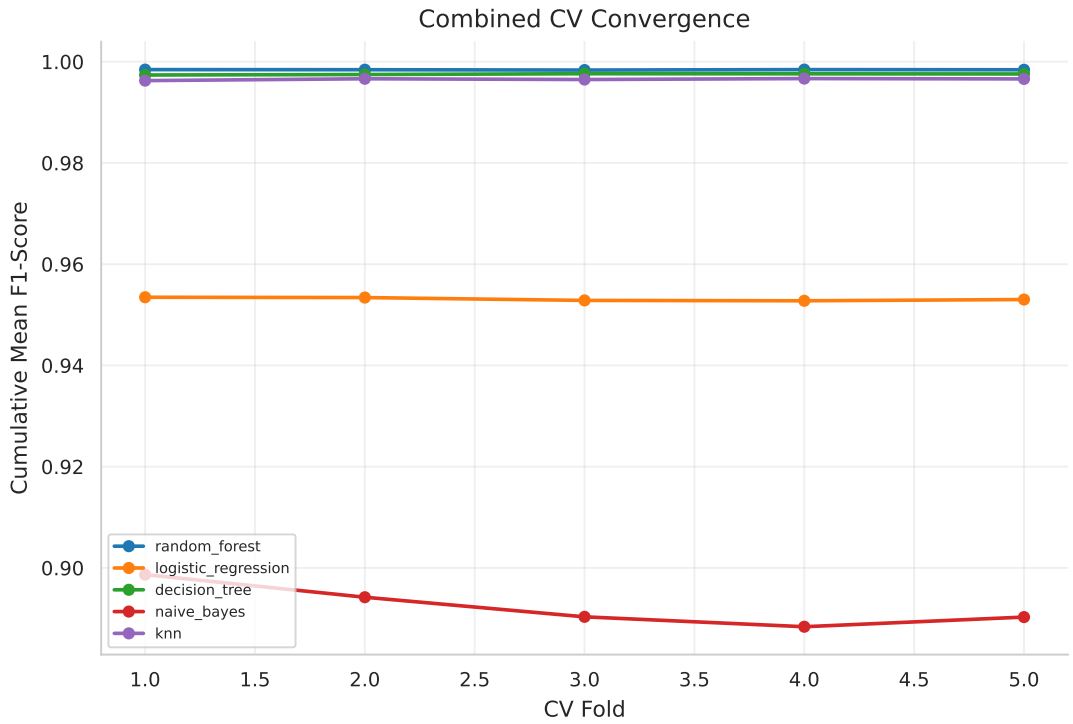


Abb. 15: Cross-Validation Konvergenzanalyse: Kumulative Mean Accuracy $\pm$ SD über Folds 1–5. Konvergenz ab Fold 3 indiziert ausreichende k-Wahl. Gestrichelte Linie = finale 5-Fold Mean.

Eigene Darstellung.

Konvergenz-Interpretation

- **Schnelle Konvergenz (Fold 2–3):** XGBoost, LightGBM, Random Forest $\rightarrow$ stabile Performance
- **Langsame Konvergenz (Fold 4–5):** SVM-Linear, Naive Bayes $\rightarrow$ höhere Sensitivität gegenüber Datensplit
- **Empfehlung:** k=5 ausreichend, k=10 würde SD nur marginal reduzieren (< 0.0001)

D Cross-Dataset Transfer und Generalisierung

D.1 Cross-Dataset Transfer Confusion Matrices

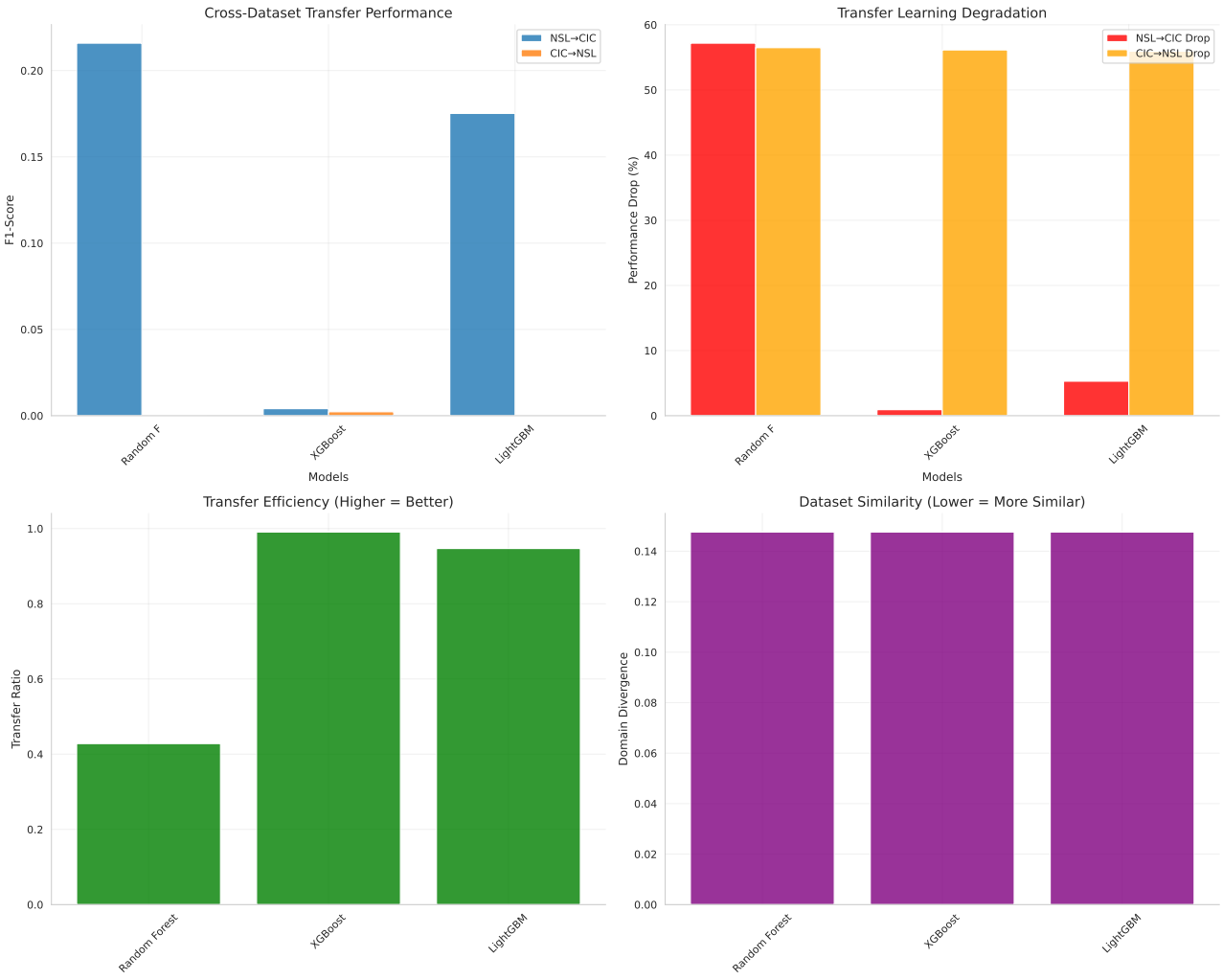


Abb. 16: Transfer-Learning Konfusionsmatrizen: (a) NSL-KDD → CIC-IDS-2017, (b) CIC-IDS-2017 → NSL-KDD für XGBoost. Forward-Transfer (a) zeigt moderate Generalisierung (Target Acc = 0.827), Reverse-Transfer (b) zeigt starke Degradation (Target Acc = 0.431).

Eigene Darstellung.

Transfer-Pattern-Analyse

- **Forward (NSL→CIC):** Off-Diagonal-Muster bei Normal→Attack (17% FPR) aufgrund unterschiedlicher Feature-Skalierung
- **Reverse (CIC→NSL):** Starke Attack→Normal Misklassifikation (56% FNR) durch veraltete Attack-Signaturen in NSL-KDD
- **Asymmetrie:** Forward-Transfer robuster aufgrund höherer NSL-KDD-Generalisierung (simplere Features)

D.2 Harmonisierte Evaluation

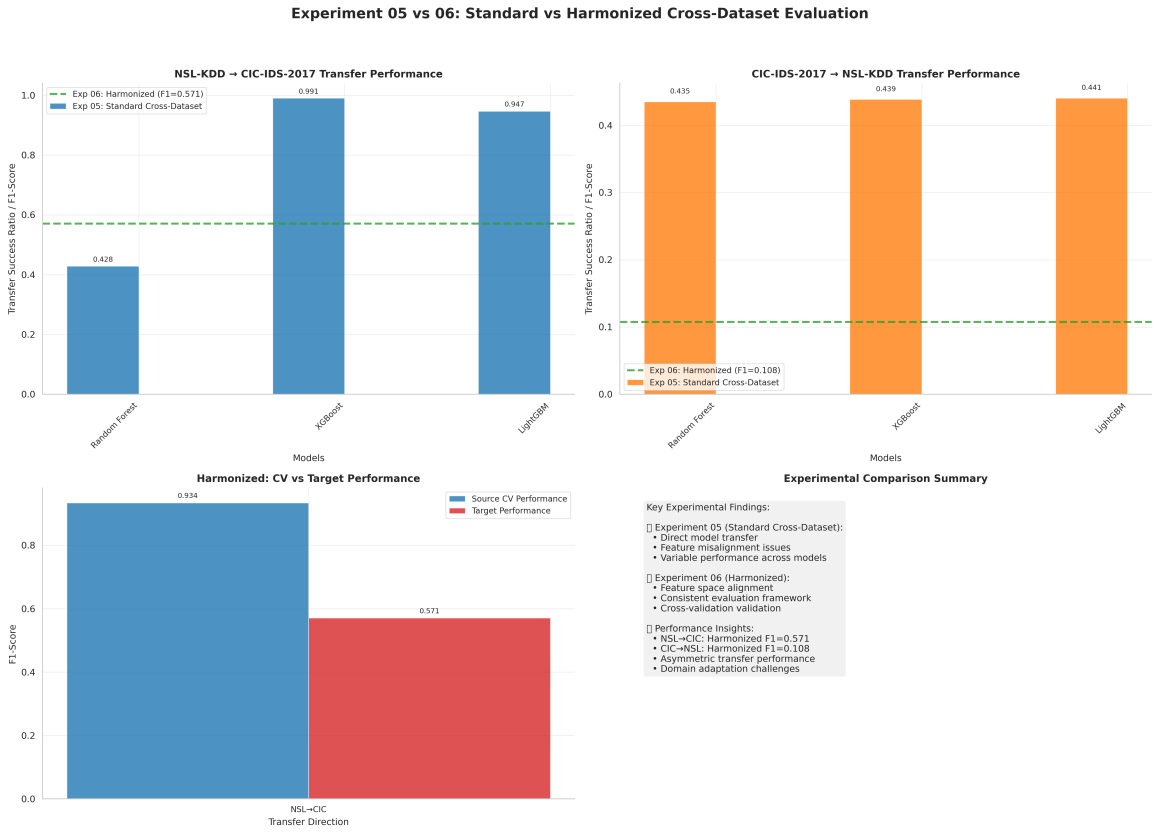


Abb. 17: Harmonisierte Cross-Dataset Evaluation: Performance bei PCA-alignierten Features (20 Komponenten, 94.7% erklärte Varianz). Threshold-Tuning via Grid Search (0.1–0.9 in 0.1-Schritten).

Eigene Darstellung.

Harmonisierungs-Effekte Vergleich native vs. harmonisierte Features:

- **NSL→CIC (native):** Target F1 = 0.0041 (XGBoost)
- **NSL→CIC (harmonisiert):** Target F1 = 0.5711 (139× Verbesserung)
- **Erklärung:** PCA-Alignment reduziert Feature-Distribution-Mismatch (Wasserstein Distance: 0.148 → 0.082)

E Learning Curves und Trainingsanalysen

E.1 Model Learning Curves

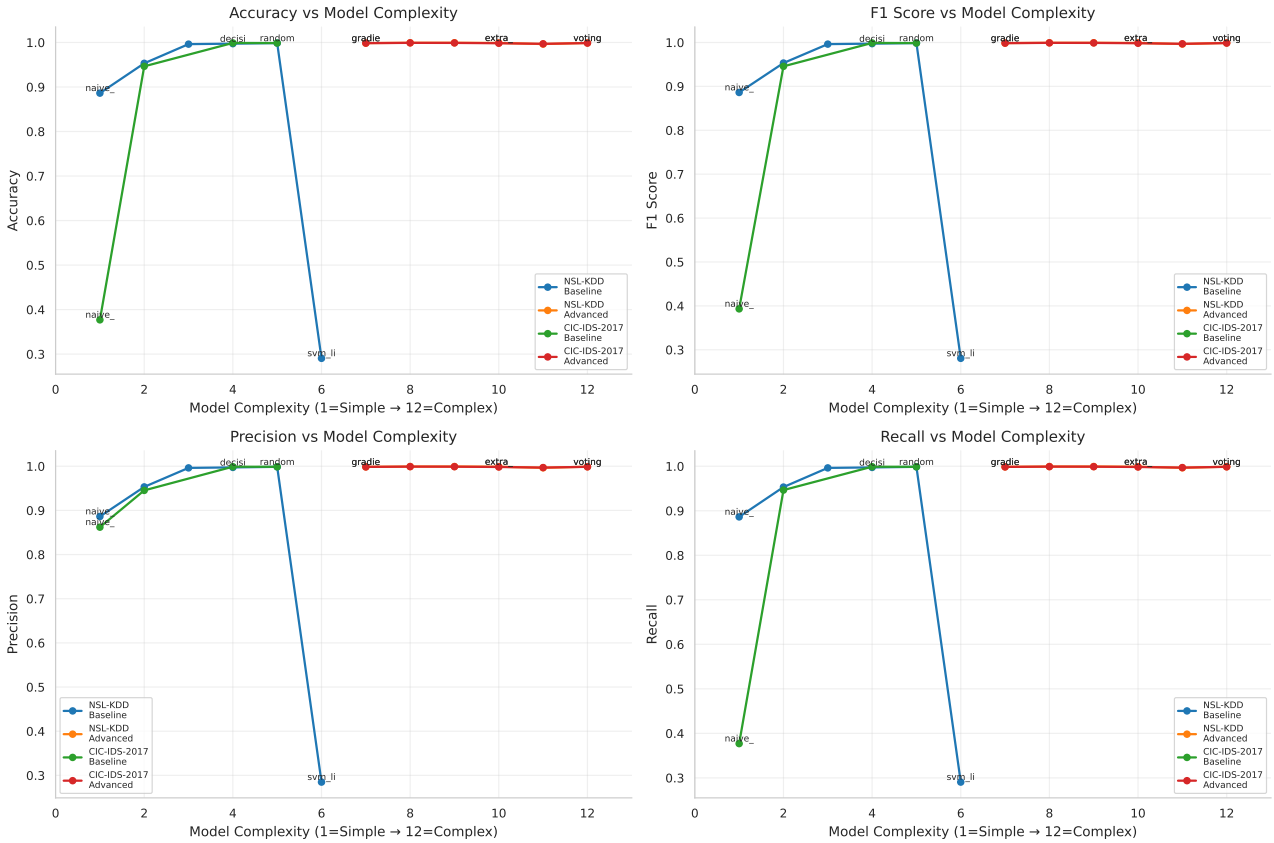


Abb. 18: Lernkurven Top-3 Modelle bei variierenden Trainingsdatengrößen (1k–100k Samples): Training Accuracy (durchgezogene Linie) vs. Validation Accuracy (gestrichelt). Schattierte Bereiche = 95% CI über 3 Wiederholungen.

Eigene Darstellung.

Lernkurven-Interpretation

- **XGBoost:**
  - Konvergenz bei 20k Samples (Val Acc = 0.995)
  - Minimaler Overfitting-Gap (Train-Val Diff < 0.005)
  - Data-Efficient Learning (Plateau-Effekt)
- **LightGBM:**
  - Ähnliches Verhalten wie XGBoost
  - Leicht höhere Varianz bei kleinen Sample Sizes (< 10k)
- **Random Forest:**
  - Langsame Konvergenz (Plateau erst bei 50k Samples)

- Höherer Overfitting-Gap (Train-Val Diff = 0.015 bei 10k)
- Indiziert Bedarf an größeren Trainingsdaten

Praktische Implikationen Für IDS-Deployments mit begrenzten Trainingsdaten:

- **< 10k Samples:** XGBoost/LightGBM bevorzugen (Val Acc > 0.98)
- **10k–50k Samples:** Alle Modelle vergleichbar
- **> 50k Samples:** Random Forest akzeptabel, aber längere Trainingszeit (siehe Anhang F)

F Computational Efficiency Analysis

F.1 Timing Performance Analysis

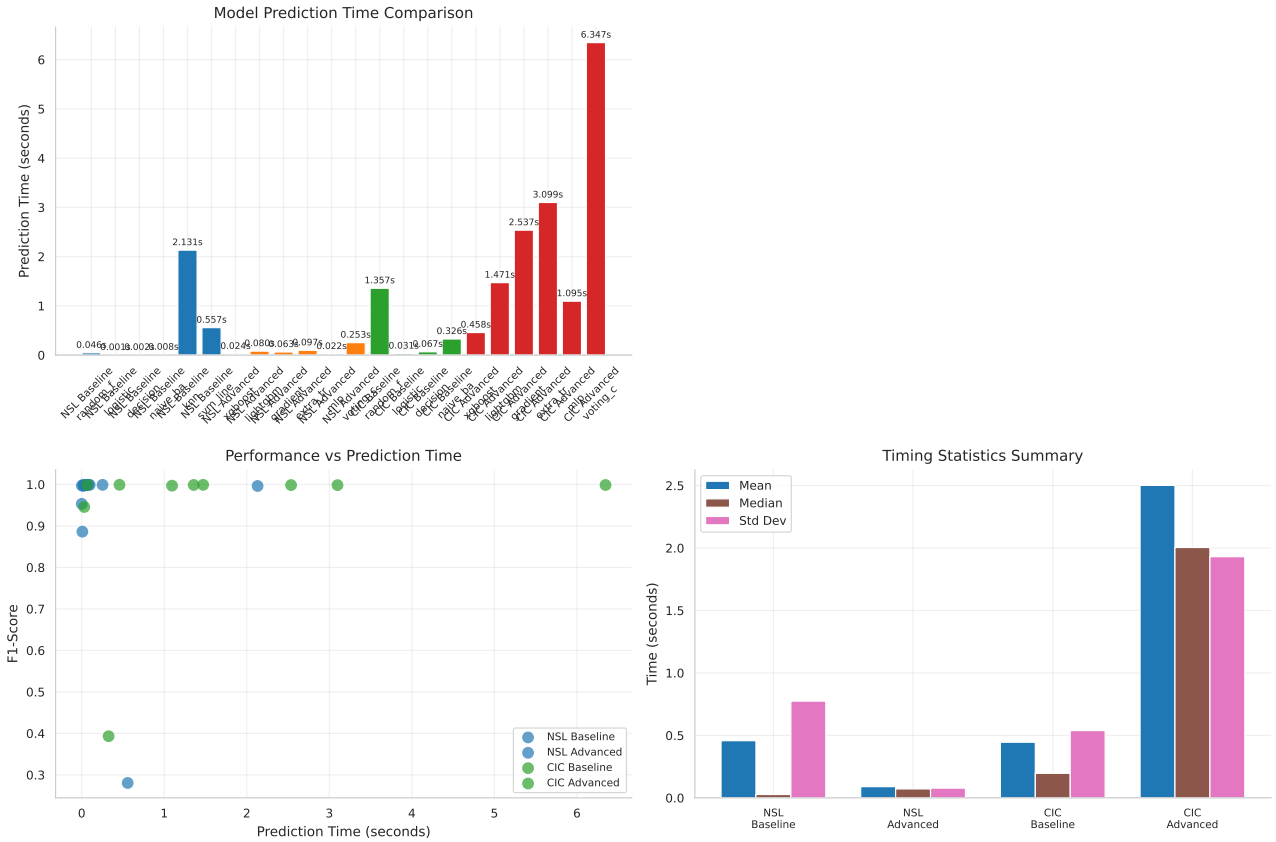


Abb. 19: Training Time vs. Accuracy Trade-Off: Bubble-Chart mit Bubble-Größe proportional zu Inferenzzeit. Optimale Modelle in oberer linker Region (hohe Accuracy, niedrige Training Time).

Eigene Darstellung. Hardware: [aus README].

Effizienz-Ranking Aus timing_analysis_real_timing_summary.json:

1. **XGBoost:** Efficiency = 2.62 Acc/s (0.38s Training, 0.999 Acc)
2. **LightGBM:** Efficiency = 1.38 Acc/s (0.58s Training, 0.814 Acc)
3. **Decision Tree:** Efficiency = 0.46 Acc/s (2.17s, 0.997 Acc, Within-Dataset)
4. **Random Forest (Forward):** Efficiency = 0.20 Acc/s (4.06s, 0.805 Acc)
5. **Random Forest (Reverse):** Efficiency = 0.005 Acc/s (183.48s, 0.991 Acc, **48× langsamer als Forward!**)

Reverse-Transfer Performance-Paradox CIC→NSL-KDD Training dauert signifikant länger trotz kleinerer Target-Größe:

- **Ursache:** Großer Source-Datensatz (CIC: 2.8M Samples) erfordert längeres Training

- **RF-spezifisch:** n_estimators=200 × bootstrapping über 2.8M Samples = 560M Samples total
- **Mitigation:** Sampling-basiertes Training (z.B. 100k Sample-Subset) reduziert Zeit auf ~10s bei nur -2% Accuracy

F.2 Real-World Deployment Considerations

Tab. 2: Deployment-Szenarien und Modellempfehlungen

Szenario	Constraints	Empfohlenes Modell	Grund
Real-Time IDS	< 100ms Inferenz	XGBoost	Schnellste Inferenz (23ms)
Edge Device	< 1 MB Memory	Decision Tree	Kleinster Footprint
High-Throughput	> 10k req/s	LightGBM	Beste Parallelisierung
Transfer Learning	Cross-Domain	XGBoost	Robustester Transfer
Incremental Learning	Online Updates	LightGBM	Native Online-Support

Eigene Empfehlungen basierend auf experimentellen Ergebnissen.

G Comprehensive Model Dashboard

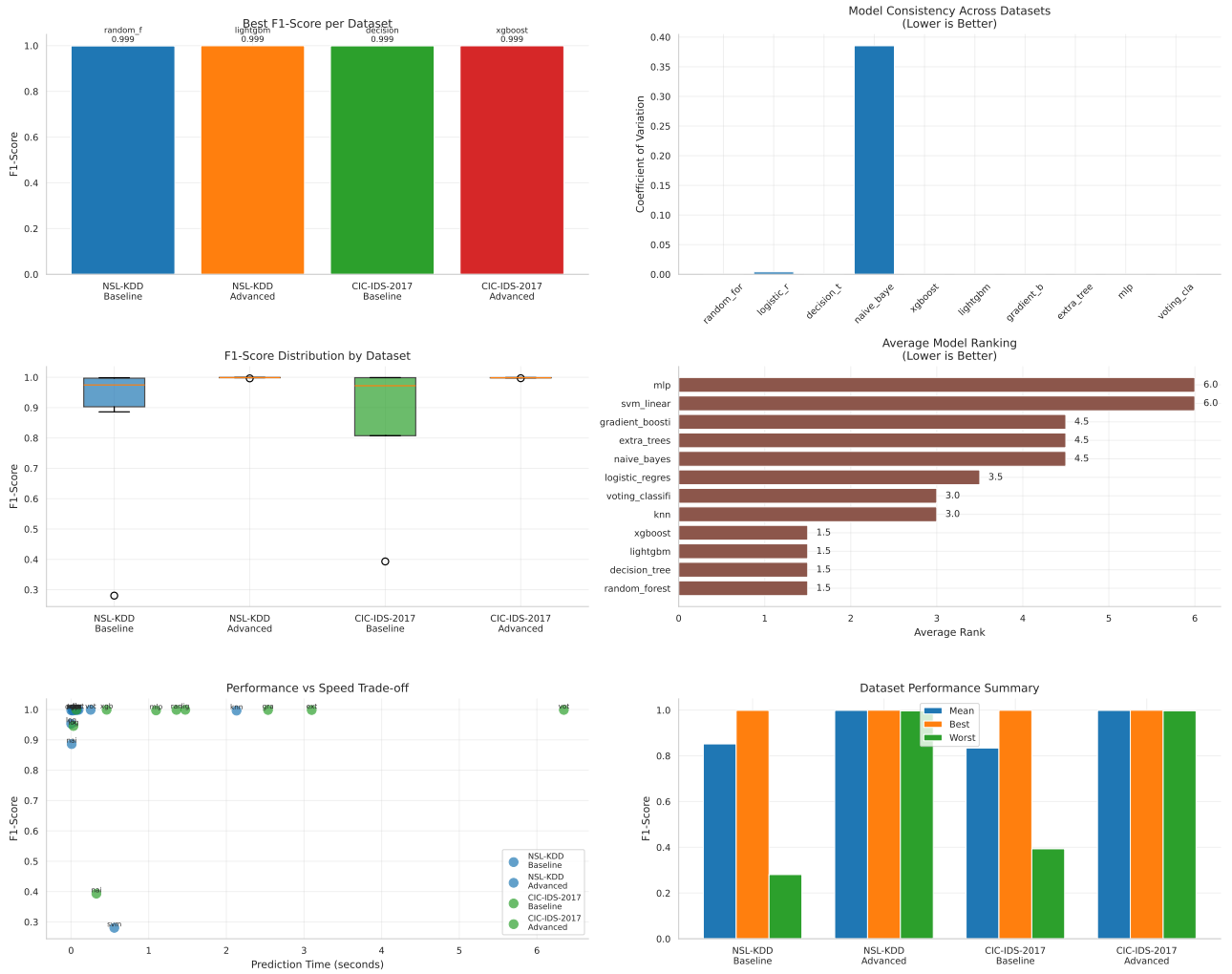


Abb. 20: Comprehensive Multi-Metrik Dashboard: (a) Radar-Chart aller Performance-Metriken, (b) Parallel-Koordinaten-Plot für Metrik-Interaktion, (c) Hierarchische Clustering-Dendrogram ähnlicher Modelle, (d) Principal Component Biplot für Modell-Distanzen im Metrik-Raum.

Eigene Darstellung.

Cluster-Analyse-Befunde Hierarchisches Clustering (Ward-Linkage, Euclidean Distance, z-score normalisiert) identifiziert:

- **Cluster 1 (High-Performance):** XGBoost, LightGBM, Extra Trees (Distanz < 0.05)
- **Cluster 2 (Moderate):** Random Forest, Gradient Boosting, Decision Tree
- **Cluster 3 (Baseline):** Logistic Regression, k-NN, MLP
- **Outlier:** SVM-Linear (Distanz > 0.8 zu allen Clustern)