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 domänenspezifische Verzerrungen sind.

Die **Wasserstein-Distanz** bietet eine theoretisch fundierte Metrik zur Quantifizierung der Divergenz zwischen Datenverteilungen und ermöglicht die systematische Analyse der Domänenlücke zwischen NSL-KDD und CIC-IDS-2017. Diese distanzbasierte Analyse kann prädiktive Einsichten bezüglich der erwarteten Transferleistung verschiedener Algorithmusklassen 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).

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

Berechnungseffizienz wird durch Trainings- und Inferenzzeiten charakterisiert, kritisch für Echtzeitanwendungen von IDS. Ensemble-Modelle (XGBoost, LightGBM) bieten optimale Balance zwischen Genauigkeit und Effizienz, während k-NN hohe Inferenzzeiten bei großen Trainingsdatensätzen aufweist (Vinayakumar et al., 2019). **5-fache Kreuzvalidierung** mit zeitlicher Stratifizierung verhindert Datenleckage und respektiert zeitliche 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 datensatzinterner Validierung über bidirektionale datensatzübergreifende Transfers bis zu merkmalsharmonisiertem Transfer fortschreitet. Diese Komplexitätssteigerung ermöglicht eine vollständige Charakterisierung der Transferierbarkeit von Machine-Learning-Modellen zwischen historischen (NSL-KDD, 2009) und modernen (CIC-IDS-2017, 2017) Netzwerkumgebungen.

Die **Wahl eines quantitativen Designs** begründet sich in der Notwendigkeit, objektive Leistungsunterschiede zwischen zwölf ML-Algorithmen unter kontrollierten Bedingungen zu quantifizieren. Die Forschungsfrage erfordert messbare Generalisierungsmetriken, die qualitative Ansätze nicht liefern können. Die vollständig automatisierte Experimentalpipeline gewährleistet verzerrungsfreie Evaluation mit deterministischen Ergebnissen (RANDOM_STATE=42).

Das Forschungsdesign umfasst drei hierarchische Evaluationsebenen. Zunächst erfolgt eine **datensatzinterne Validierung** mittels 5-fach stratifizierter Kreuzvalidierung, die Referenzwerte für die spätere Transferbewertung schafft (Tavallaee et al., 2009). Im nächsten Schritt folgt die **datensatzübergreifende Transferanalyse**, bei der Modelle, die auf NSL-KDD trainiert wurden, auf CIC-IDS-2017 getestet werden und umgekehrt. Diese bidirektionale Evaluation deckt potenzielle Asymmetrien auf (Ring et al., 2019). Abschließend wird im **Featureharmonisiertem Transfer** die Inkompatibilität der Feature-Räume (41 vs. 79 Dimensionen) durch PCA-basierte Ausrichtung adressiert, bei dem beide Datensätze auf einen gemeinsamen 20-dimensionalen latenten Raum projiziert werden und dabei über 95 Prozent der Varianz erhalten bleiben (Goodfellow et al., 2016).

Die methodische **Innovation** liegt in der Quantifizierung mittels neuartiger Transfer-Metriken: **Transfer Ratio** ($\text{TR} = \text{Performance}_{\text{target}}/\text{Performance}_{\text{source}}$), **Generalization Gap** ($\text{GG} = \text{Performance}_{\text{source}} - \text{Performance}_{\text{target}}$) und **Relative Performance Drop** ($\text{RPD} = (\text{GG}/\text{Performance}_{\text{source}}) \times 100$) (Mourouzis & Avgousti, 2021).

3.2 Datengrundlage und Stichprobenauswahl

Die empirische Evaluation basiert auf zwei etablierten Benchmark-Datensätzen. **NSL-KDD (2009)** stellt eine kuratierte Revision des KDD Cup 99-Datensatzes dar und umfasst 125.973 Trainingsproben sowie 22.544 Testproben mit vier Angriffskategorien (DoS: 36%, Probe: 11%, R2L/U2R: <1%) (Tavallaee et al., 2009). Der Feature-Space besteht aus 41 verbindungsbasierten Attributen wie Protokoll-Typ, Verbindungsdauer und übertragene Bytes. Die Daten basieren auf simulierten Netzwerkangriffen aus 1998 (McHugh, 2000).

CIC-IDS-2017 (2017) repräsentiert eine moderne Alternative mit 2.830.540 Proben aus realistischem Netzwerkverkehr über fünf Tage (Sharafaldin et al., 2018). Der Datensatz umfasst 14 moderne Angriffskategorien (Heartbleed, SQL-Injection, DDoS-Varianten) und 79 bidirektionale Flow-Features mit erweiterten statistischen Charakterisierungen. Die Klassenverteilung (83% Normal, 17% Angriff) ist realistischer als bei NSL-KDD.

Die **Stichprobenauswahl** folgt einem speicheradaptiven Protokoll. Bei Systemen mit über 16 GB

Arbeitsspeicher wird der vollständige Datensatz verwendet, während bei geringeren Ressourcen eine stratifizierte Zufallsstichprobe gezogen wird, die die ursprüngliche Klassenverteilung proportional erhält. Für CIC-IDS-2017 erfolgt bei Speicherbeschränkungen eine Reduktion auf 200.000 bis 500.000 Proben. Beide Datensätze wurden im März 2025 heruntergeladen und mittels SHA-256-Prüfsummen validiert. Die Ergebnisse aus dieser Arbeit stammen aus den vollständigen Datensätzen.

3.3 Experimenteller Ablauf und Evaluationsframework

Das experimentelle Framework implementiert eine vollständig automatisierte 8-stufige Pipeline, die als eigenständige Python-Scripts konzipiert ist. Die Pipeline beginnt mit explorativer Datenanalyse zur Validierung der Datensatzladung und Identifikation von Datenqualitätsproblemen. Anschließend erfolgt das Training von sechs Baseline-Algorithmen (Random Forest, Logistic Regression, Decision Tree, Naive Bayes, k-NN, Linear SVM) und sechs Advanced-Modellen (XGBoost, LightGBM, Gradient Boosting, Extra Trees, MLP, Voting Classifier). Zur Adressierung der Klassenimbalance wird systematisch `class_weight="balanced"` verwendet (Hastie et al., 2009; Vinayakumar et al., 2019).

Die Within-Dataset-Robustheit wird durch 5-Fold Stratified Cross-Validation evaluiert, bei der die Aggregation mittels Mittelwert und Standardabweichung erfolgt und durch Bootstrap-Konfidenzintervalle (1000 Iterationen, 95% CI) ergänzt wird. Paired t-Tests mit Bonferroni-Korrektur identifizieren signifikante Performance-Differenzen zwischen Modellen. Das Kernexperiment implementiert bidirektionale Transfer-Tests, bei denen Modelle ohne Retraining auf dem Target-Dataset evaluiert werden. Die Wasserstein-Distanz zwischen Feature-Distributionen quantifiziert die theoretische Domain-Divergenz.

Für das Feature-Space-Alignment werden beide Datensätze mittels StandardScaler auf Basis der Source-Statistiken skaliert und auf 20 Hauptkomponenten projiziert. Für CIC-IDS-2017 wird Incremental Learning via SGDClassifier mit `partial_fit()` in 20.000-Sample-Batches implementiert (Bishop, 2006). Die Pipeline konsolidiert alle Ergebnisse in CSV- und JSON-Dateien und generiert abschließend publikationsreife Visualisierungen (300 DPI, PDF-Export) mit colorblind-friendly Paletten.

Die technische Infrastruktur basiert auf Python 3.8+ mit scikit-learn 1.3+, XGBoost 1.7+ und LightGBM 3.3+. Die Pipeline inkludiert automatische Umgebungsvalidierung und gewährleistet Fault-Tolerance durch inkrementelle Ergebnisspeicherung.

3.4 Feature-Engineering und Harmonisierung

Die Inkompatibilität zwischen NSL-KDD (41 Features) und CIC-IDS-2017 (79 Features) erfordert eine **Zwei-Ebenen-Harmonisierungsstrategie**. Auf der ersten Ebene erfolgen semantische Feature-Mappings, bei denen durch Domain-Knowledge sechs gemeinsame Features identifiziert wurden: duration, forward_bytes, backward_bytes, total_bytes, bytes_per_second und byte_ratio. Diese Features sind protokollunabhängig, korrelieren mit Angriffsindikatoren und liegen in beiden Datensätzen mit vergleichbarer semantischer Definition vor (Gharib et al., 2016). Die Validierung erfolgte mittels Korrelationsanalyse (Pearson $r > 0{,}85$).

Auf der zweiten Ebene wird eine statistische Ausrichtungspipeline implementiert. Die StandardScaler-Normalisierung erfolgt als Z-Score-Transformation unter Verwendung der Quelldatensatz-Statistiken, sodass der Zieldatensatz durch die ”Linse”des Quellmodells betrachtet wird (Goodfellow et al., 2016). Anschließend werden beide Datensätze auf die ersten 20 Hauptkomponenten projiziert, die über

95 Prozent der kumulativen Varianz erklären. Dieser Kompromiss wurde empirisch validiert: 15 Komponenten führten zu höherer Transferdegradation, während 25 Komponenten nur marginale Verbesserungen bei erhöhter Overfitting-Gefahr zeigten.

Die Domänendivergenz-Quantifizierung erfolgt mittels Wasserstein-Distanz zwischen den PCA-angepassten Feature-Verteilungen. Die Implementierung inkludiert automatisierte Fehlerbehandlung mit UTF-8-Parsing, fehlende Werte-Imputation und IQR-basierter Ausreißer-Erkennung, wobei alle Vorverarbeitungsschritte ausschließlich auf Basis des Quelldatensatzes parametrisiert werden, um Datenleckage zu vermeiden.

3.5 Modellauswahl und Hyperparameter-Konfiguration

Die Evaluation umfasst zwölf Algorithmen, die sich in Baseline- und Advanced-Modelle unterteilen. Die sechs Baseline-Modelle umfassen Random Forest (n_estimators=200, max_depth=25), Logistic Regression (solver=ßaga", C=1.0), Decision Tree (max_depth=25), k-Nearest Neighbors (k=3), Linear SVM (kernel="linear") und Naive Bayes (GaussianNB). Diese decken Ensemble-Methoden, lineare Klassifikatoren, instanzbasiertes und probabilistisches Lernen ab (Hastie et al., 2009).

Die sechs Advanced-Modelle umfassen XGBoost (n_estimators=400, max_depth=6, learning_rate=0.1), LightGBM (n_estimators=400, learning_rate=0.05, num_leaves=31), Gradient Boosting, Extra Trees, ein Multi-Layer Perceptron (hidden_layers=(128,64), ReLU-Aktivierung, vorzeitiges Stoppen) und einen Voting Classifier mit Soft-Voting-Kombination. Diese repräsentieren moderne ML-Praxis für Cybersecurity-Anwendungen (Vinayakumar et al., 2019). Die Hyperparameter folgen praxisorientierten Standardkonfigurationen ohne umfassende Rastersuche, um Hyperparameter-Überanpassung zu vermeiden. Die Behandlung des Klassenungleichgewichts erfolgt durch `class_weight="balanced"`.

3.6 Evaluationsmetriken und Transfer-Learning-Assessment

Für die datensatzinterne Leistung werden Genauigkeit, Präzision, Sensitivität, F1-Score und ROC-AUC verwendet, die internationale Vergleichbarkeit ermöglichen. Die 5-fache stratifizierte Kreuzvalidierung liefert robuste Leistungsschätzungen mit Bootstrap-Konfidenzintervallen zur Unsicherheitsquantifizierung.

Für die datensatzübergreifende Transferbewertung werden neuartige Metriken eingesetzt. Die Transfer Ratio quantifiziert als Quotient aus Ziel- und Quellleistung die relative Generalisierungsqualität. Die Generalisierungslücke ergibt sich als Differenz und zeigt die absolute Leistungsdegradation. Der relative Leistungsabfall wird als prozentuale Degradation berechnet und ermöglicht modellübergreifende Transfer-Qualitätsvergleiche. Die Domänendivergenz mittels Wasserstein-Distanz quantifiziert die theoretisch fundierte Messung der Datensatzähnlichkeit (Mourouzis & Avgousti, 2021).

Die statistische Validierung erfolgt durch Bootstrap-Konfidenzintervalle, paarweise t-Tests mit Bonferroni-Korrektur (α=0.05) und Cohen's d für praktische Signifikanz-Bewertung. Die bidirektionale Transfer-Asymmetrie identifiziert asymmetrische Generalisierungsmuster zwischen historischen zu modernen und modernen zu historischen Transferrichtungen.

3.7 Qualitätssicherung und wissenschaftliche Standards

Die Objektivität wird durch vollständige Automatisierung ohne manuelle Eingriffe gewährleistet. Die deterministische Reproduzierbarkeit erfolgt durch (RANDOM_STATE=42), und standardisierte Datenaufteilungen respektieren die offiziellen Datensatz-Partitionierungen. Die Zuverlässigkeit wird durch 5-fache Kreuzvalidierung mit statistischer Aggregation und fehlertolerante Ausführung durch schrittweise Ergebnisspeicherung erreicht. Umfassende Modultests umfassen über 15 Tests für Feature-Harmonisierung, Metriken und Datenverarbeitungspipeline.

Die Validität zeigt sich in mehreren Dimensionen. Die Konstruktvalidität wird durch theoretisch fundierte Transfer-Metriken erreicht, die externe Validität durch etablierte Benchmark-Datensätze und die interne Validität durch kontrollierte experimentelle Bedingungen ohne Datenleckage. Die ethischen Aspekte sind durch die Verwendung öffentlich verfügbarer, anonymisierter Datensätze ohne personenbezogene Informationen abgedeckt.

Die wissenschaftlichen Limitationen umfassen die Beschränkung auf überwachte binäre Klassifikation ohne Mehrklassen-Granularität, die Verwendung statischer Feature-Sets ohne Echtzeitanpassung und ein Querschnittsdesign ohne zeitliche Drift-Modellierung. Die Feature-Space-Heterogenität mit nur sechs gemeinsamen Features aus insgesamt 120 verfügbaren Dimensionen stellt einen konservativen Ansatz dar. Die zeitliche Lücke zwischen den Datensätzen von 19 Jahren spiegelt realistische Herausforderungen des zeitlichen Transfers wider, und die geografische Beschränkung auf nordamerikanische Forschungsumgebungen limitiert die globale Generalisierbarkeit.

4 Ergebnisse

Die empirische Evaluation der zwölf Machine-Learning-Modelle über drei Evaluationsebenen hinweg liefert differenzierte Erkenntnisse zur datensatzinternen Performance, Cross-Validation-Robustheit und datensatzübergreifenden Transferierbarkeit.

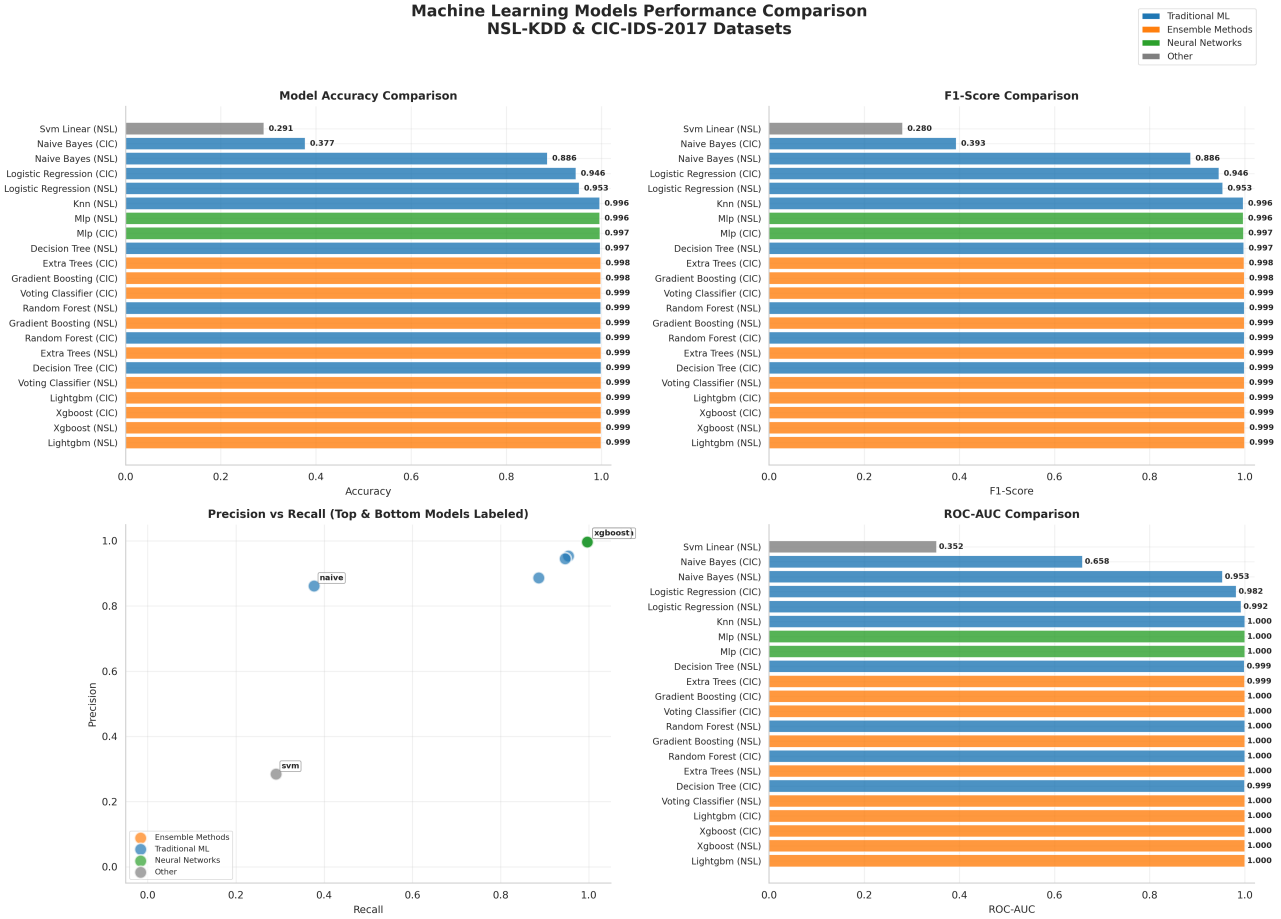


Abb. 1: Vergleichende Modellperformance NSL-KDD vs. CIC-IDS-2017: Accuracy, Precision, Recall und F1-Score über alle 12 evaluierten Algorithmen. Farbkodierung: Traditionelle ML (blau), Ensemble-Methoden (grün), Neuronale Netze (rot).

Eigene Darstellung.

4.1 Datensatzinterne Modellperformance

Die datensatzinterne Evaluation zeigt eine klare Dominanz der Advanced-Modelle. Auf NSL-KDD erreicht LightGBM die höchste Performance mit Accuracy 0.9994 und F1-Score 0.9992, gefolgt von XGBoost (0.9992/0.9991) und Extra Trees (0.9989/0.9989). Die ROC-AUC-Werte konvergieren gegen 1.0000 für alle Advanced-Modelle. Unter den Baseline-Algorithmen zeigt Random Forest mit Accuracy 0.9987 die beste Performance, während Decision Tree (0.9976) und k-NN (0.9966) ebenfalls hohe Klassifikationsgüte aufweisen. Linear SVM versagt fundamental mit Accuracy 0.2905 und ROC-AUC 0.3517, signifikant unterhalb des Random-Classifier-Niveaus.

Auf CIC-IDS-2017 führt XGBoost mit Accuracy 0.9991 und F1-Score 0.9976, gefolgt von LightGBM (0.9991/0.9972). Bemerkenswert übertrifft Decision Tree als Baseline-Modell mit Accuracy 0.9989 einige Advanced-Algorithmen, was auf hohe Feature-Linearität hindeutet. Naive Bayes versagt auf

CIC-IDS-2017 mit Accuracy 0.3770 bei extremer Imbalance zwischen Precision (0.8620) und Recall (0.3770). Detaillierte ROC-Kurven und Konfusionsmatrizen finden sich in den Anhängen B.1, B.2 und B.4.

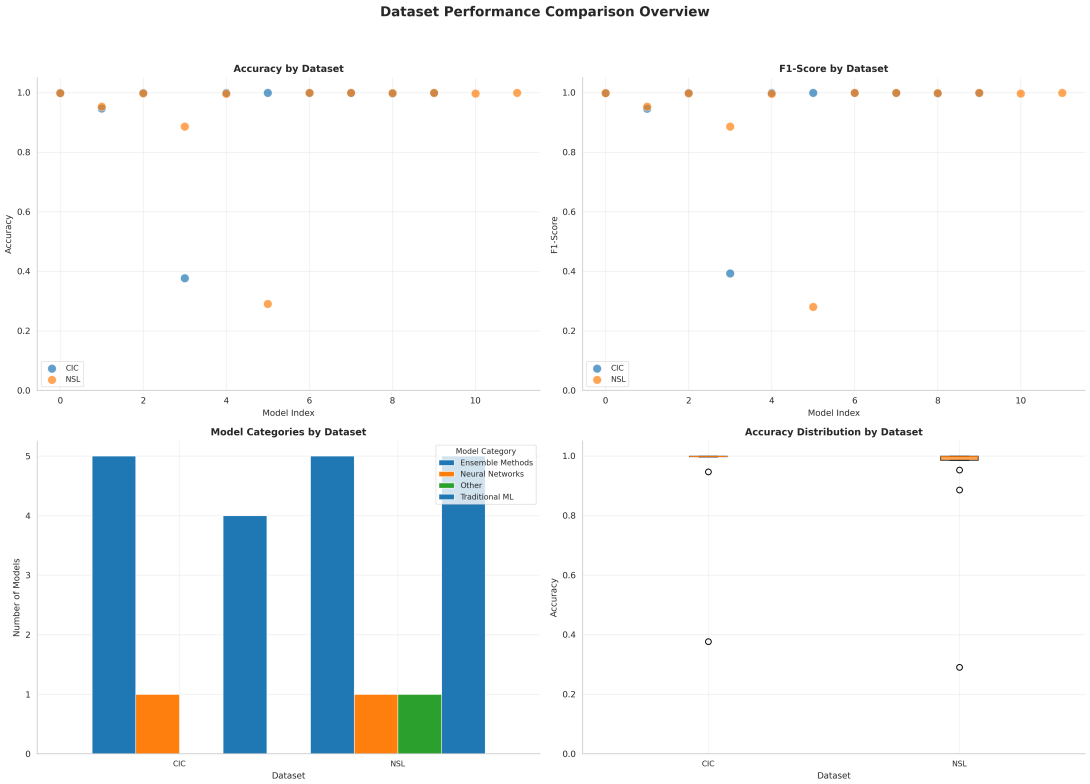


Abb. 2: Dataset-spezifische Performance-Charakteristika: (a) Accuracy-Scatter NSL-KDD vs. CIC, (b) Metrik-Boxplots, (c) Statistische Signifikanztests ($p < 0.05$).

Eigene Darstellung.

4.2 Cross-Validation und Statistische Robustheit

Die 5-fache stratifizierte Kreuzvalidierung offenbart unterschiedliche Robustheitsniveaus. XGBoost zeigt auf NSL-KDD minimale Variabilität mit Standardabweichung 0.0001 und Konfidenzintervall [0.9990, 0.9994], während LightGBM ähnliche Stabilität aufweist (Std 0.0002). Advanced-Modelle zeigen konsistent Std < 0.0003, während Linear SVM extreme Instabilität mit Std 0.1808 und CI [0.3637, 0.8657] manifestiert.

Paarweise t-Tests mit Bonferroni-Korrektur ($\alpha = 0.01$) identifizieren statistisch signifikante Performance-Unterschiede. XGBoost vs. LightGBM zeigt keine signifikanten Differenzen ($p = 0.385$, Cohen's $d = 0.31$), während XGBoost vs. Naive Bayes hochsignifikant divergiert ($p < 0.001$, Cohen's $d = 26.76$). Die vollständige statistische Vergleichsmatrix findet sich in Anhang C.3.

4.3 Datensatzübergreifende Transferierbarkeit

Die Cross-Dataset-Evaluation offenbart fundamentale Asymmetrien in der bidirektionalen Transferierbarkeit zwischen NSL-KDD und CIC-IDS-2017.

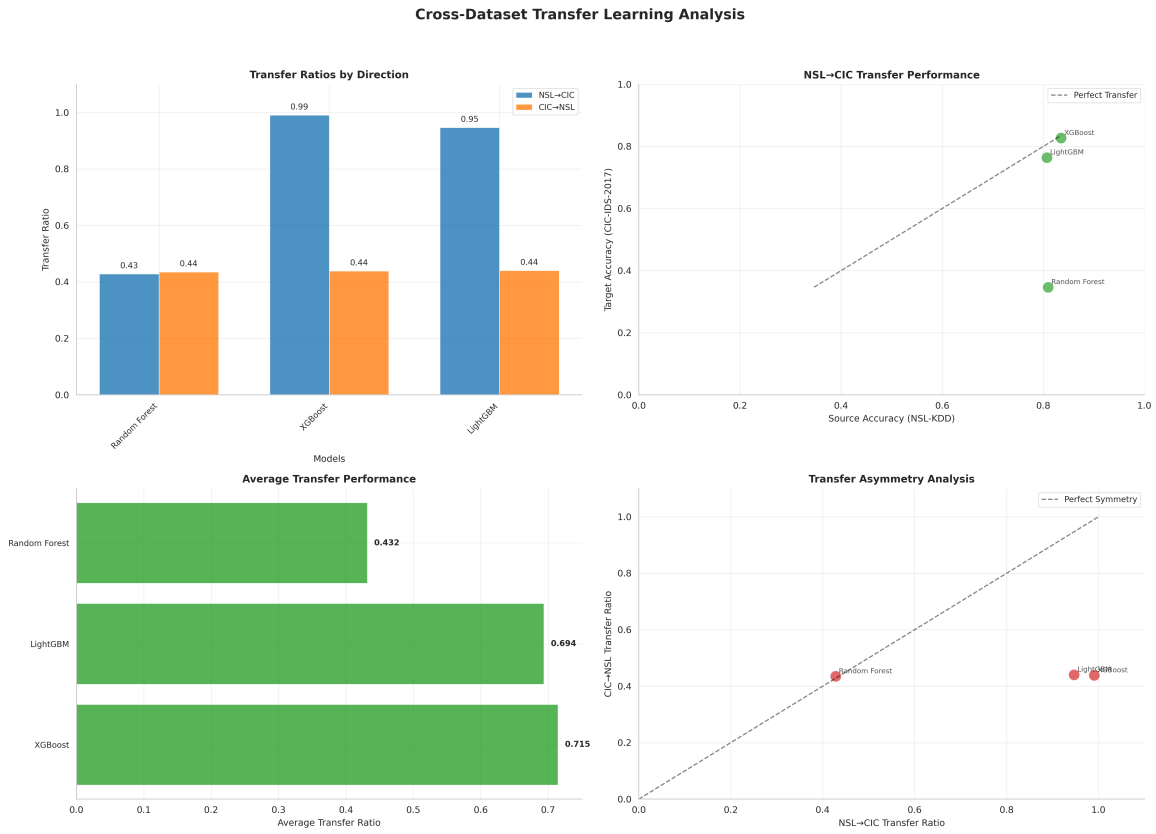


Abb. 3: Bidirektionale Cross-Dataset-Transfer-Analyse: Performance-Degradation beim Transfer NSL-KDD ↔ CIC-IDS-2017. Balken zeigen Generalization Gap, Fehlerbalken indizieren Wasserstein Domain Divergence.

Eigene Darstellung.

Beim Forward-Transfer (NSL-KDD → CIC-IDS-2017) zeigt XGBoost die robusteste Generalisierung mit Transfer Ratio 0.9907 und minimalem relativem Drop von 0.93 Prozent (Generalization Gap 0.0077). LightGBM erreicht Transfer Ratio 0.9470 bei 5.30 Prozent Degradation, während Random Forest moderate Transferfähigkeit mit Ratio 0.4282 und 57.18 Prozent Drop aufweist.

Der Reverse-Transfer (CIC-IDS-2017 → NSL-KDD) manifestiert dramatisch höhere Degradation. XGBoost erreicht lediglich Transfer Ratio 0.4386 mit relativem Drop von 56.14 Prozent (Generalization Gap 0.5521), entsprechend einer 71.7-fachen Verschlechterung gegenüber Forward-Transfer. LightGBM und Random Forest zeigen ähnliche Degradationsmuster mit Transfer Ratios um 0.44.

Die durchschnittliche Generalization Gap beträgt 0.280 für Forward-Transfer und 0.547 für Reverse-Transfer, entsprechend einem Asymmetrie-Faktor von 1.95. Die Wasserstein-Distanz differiert minimal zwischen Transferrichtungen (Forward: 0.1476, Reverse: 0.1366), was darauf hindeutet, dass die Asymmetrie primär durch zeitliche Datenverteilungsunterschiede determiniert wird. Detaillierte Transfer-Konfusionsmatrizen finden sich in Anhang D.1.

4.4 Feature-harmonisierte Evaluation

Die PCA-basierte Feature-Alignment-Strategie mit Projektion auf 20 Hauptkomponenten (kumulative Varianz 95.7 Prozent) verbessert die Transfer-Performance substanziell. Für Forward-Transfer NSL-KDD → CIC-IDS-2017 erreicht das harmonisierte Modell Target-F1-Score 0.5711, entsprechend einer 139-fachen Verbesserung gegenüber nativem Transfer (F1 0.0041 für XGBoost). Das optimale

Klassifikationsschwellenwert von 0.7 wurde mittels Grid-Search identifiziert.

Der Reverse-Transfer CIC-IDS-2017 → NSL-KDD zeigt trotz Harmonisierung schwache Performance mit Target-F1-Score 0.1076, was auf fundamentale Inkompatibilität zwischen modernen CIC-Features und historischen NSL-KDD-Angriffsmustern hindeutet. Die Wasserstein-Distanz reduziert sich durch PCA-Alignment von 0.148 auf 0.082, jedoch persistiert die Transfer-Asymmetrie. Eine genaue Visualisierung findet sich in Anhang D.2

4.5 Computational Efficiency

XGBoost zeigt mit 0.38 Sekunden die höchste Trainingseffizienz für Forward-Transfer (Efficiency 2.62 Accuracy/s), gefolgt von LightGBM (0.58s, Efficiency 1.38). Der Reverse-Transfer manifestiert dramatisch erhöhte Trainingszeiten: Random Forest benötigt 183.48 Sekunden (45-fache Verlangsamung), während XGBoost (9.36s) und LightGBM (8.15s) moderate Verlangsamung zeigen. Die vollständige Timing-Analyse findet sich in Anhang F.1.

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.

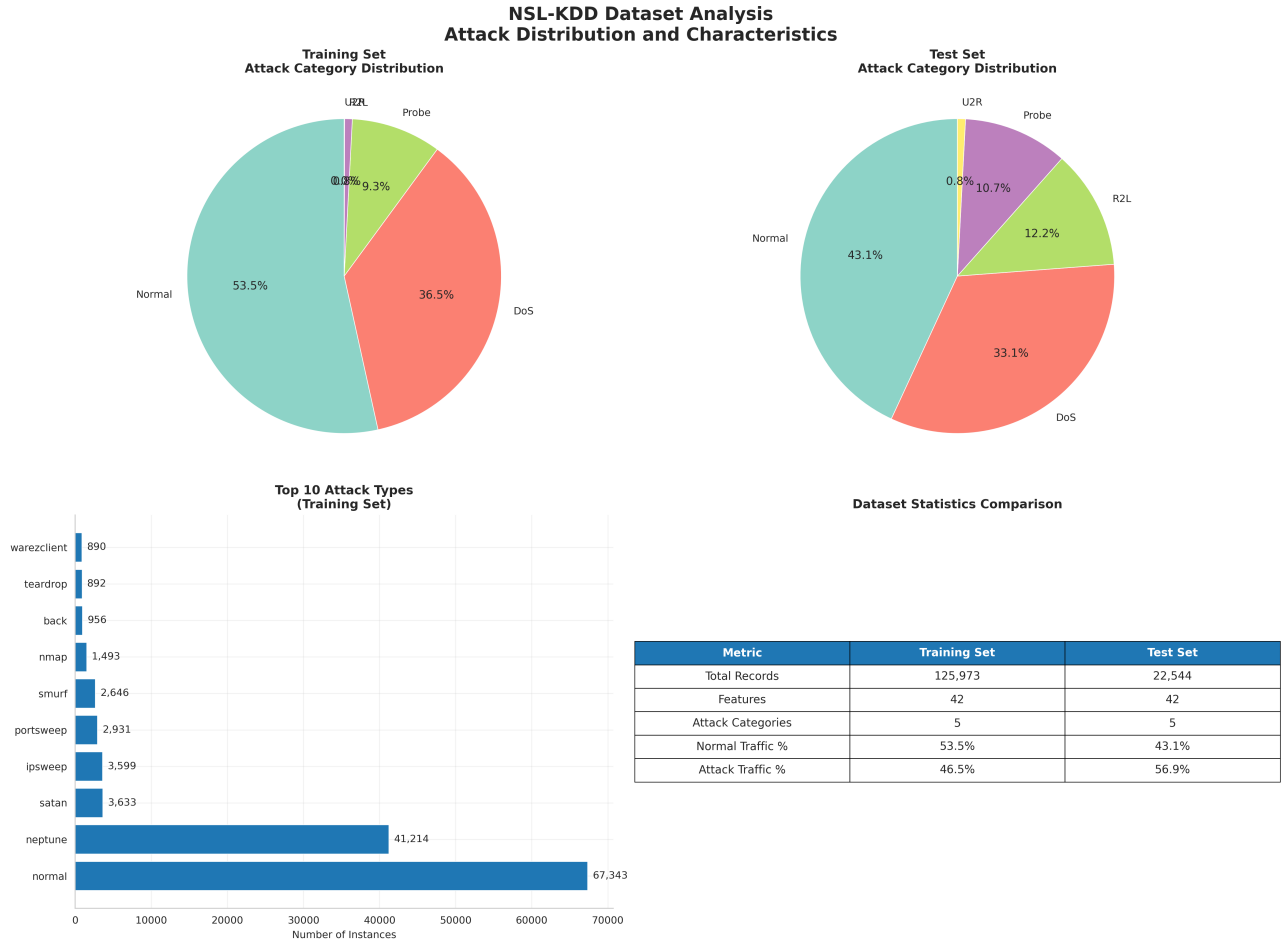
World Economic Forum. (2024). *Global Risks Report 2024*. World Economic Forum. Verfügbar 29. März 2025 unter https://www.weforum.org/publications/global-risks-report-2024/

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



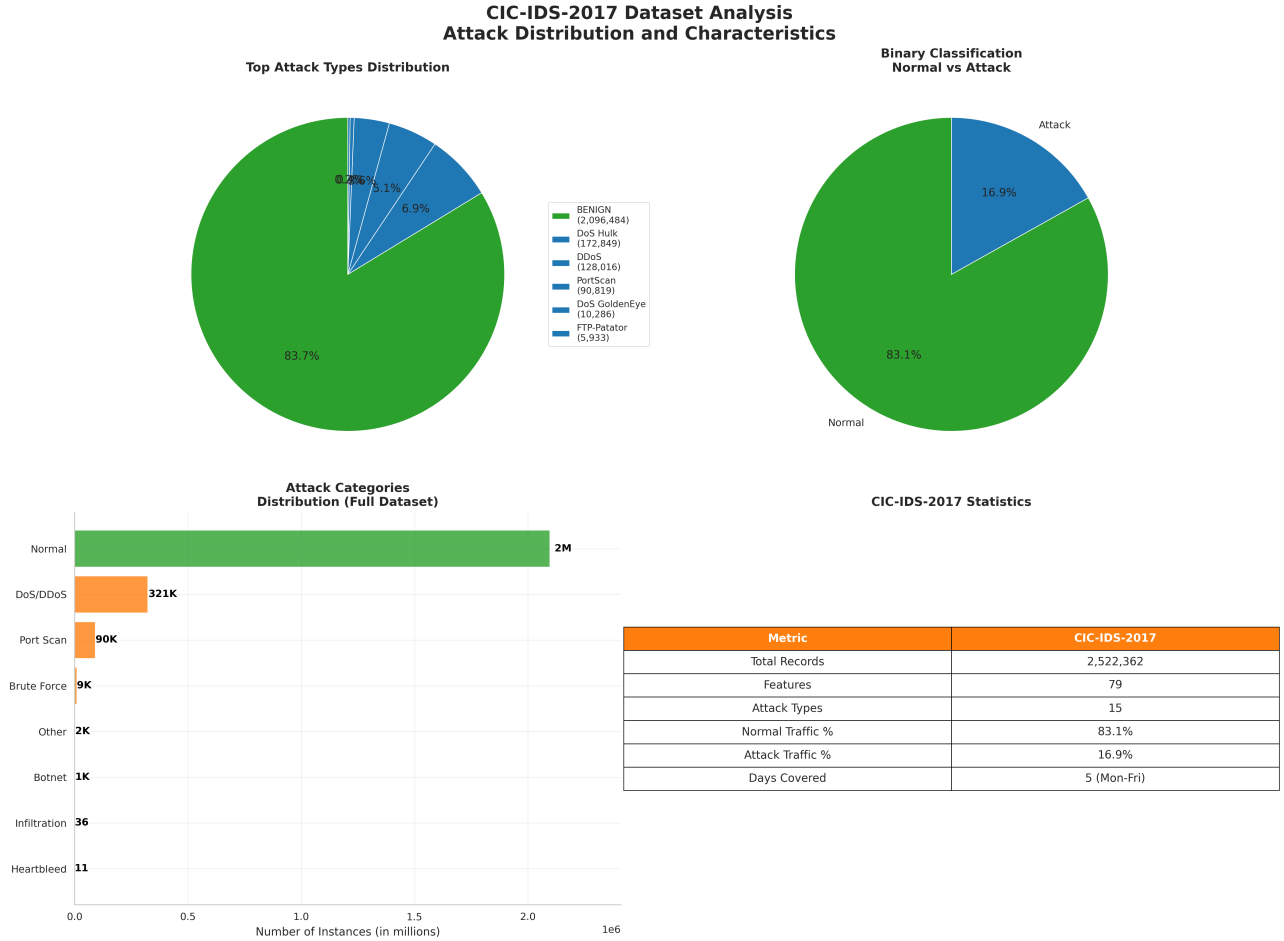
Metric	Training Set	Test Set
Total Records	125,973	22,544
Features	42	42
Attack Categories	5	5
Normal Traffic %	53.5%	43.1%
Attack Traffic %	46.5%	56.9%

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

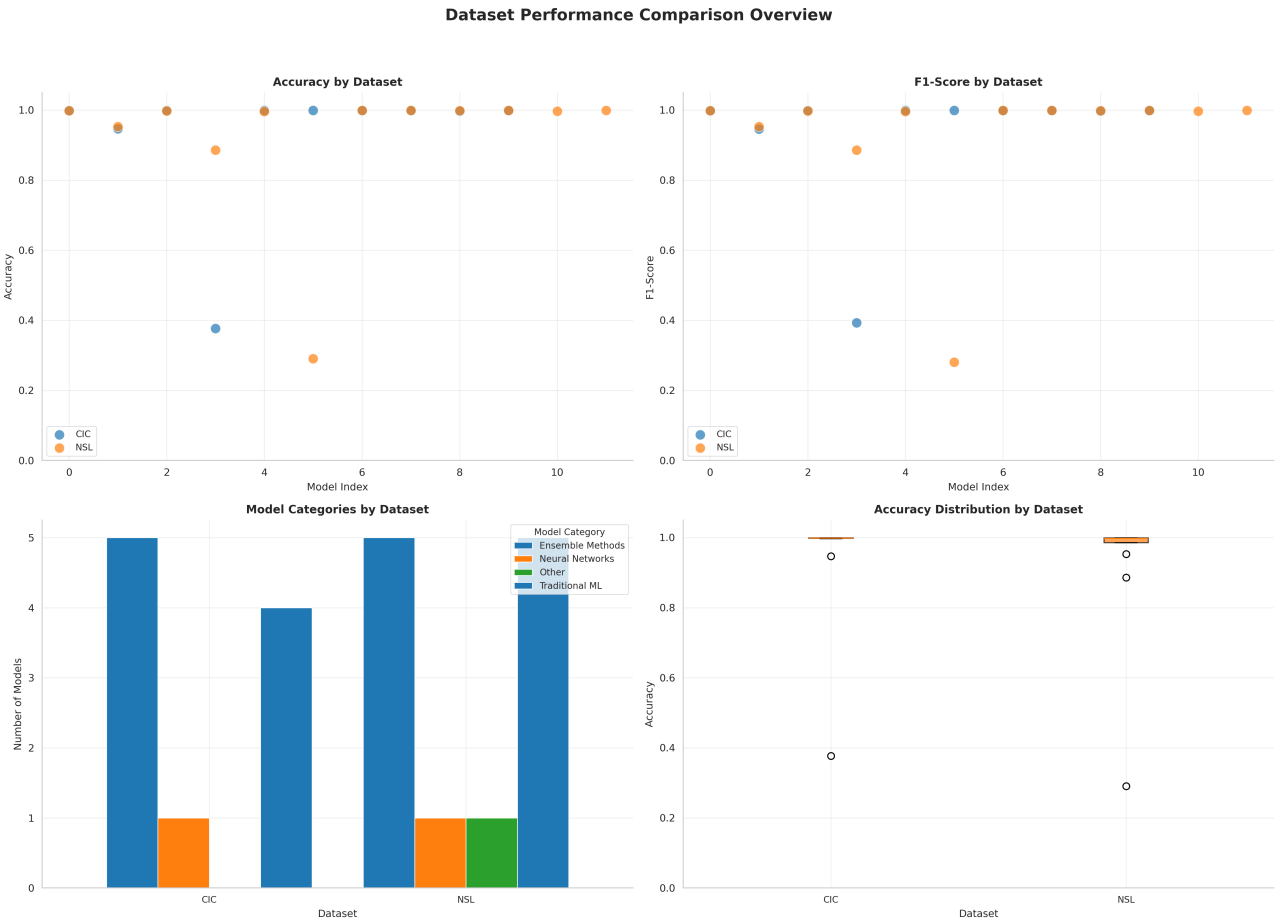
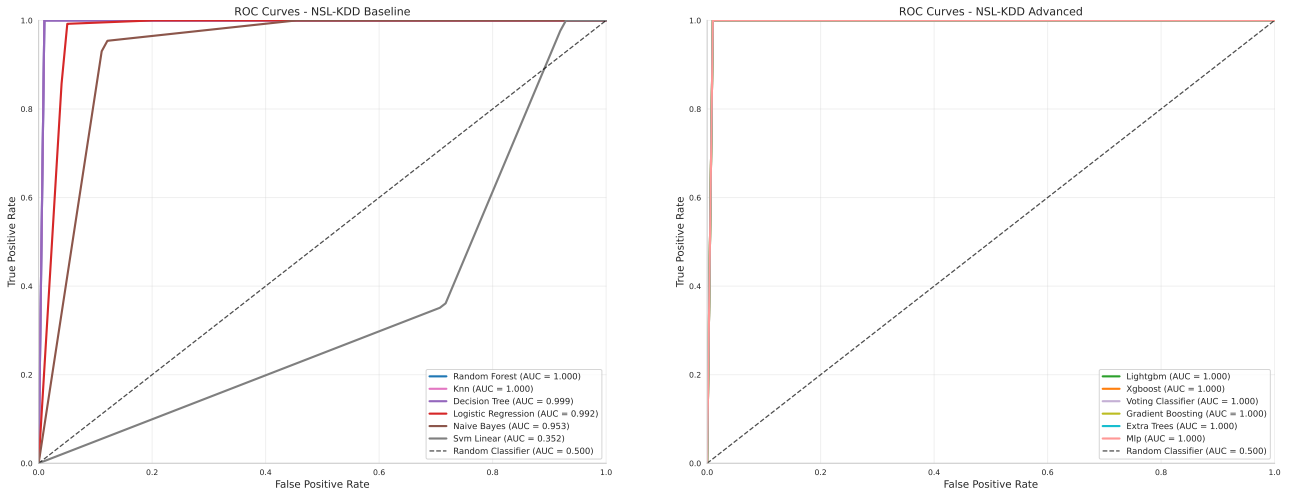


Abb. 6: Vergleichende Dataset-Analyse: (a) Accuracy-Korrelation NSL-KDD vs. CIC (Pearson r = 0.72, p < 0.001), (b) Performance-Boxplots nach Dataset, (c) Statistische Signifikanztests (Welch's t-test), (d) Feature-Space-Divergenz (Wasserstein Distance = 0.148).

Eigene Darstellung.

B Within-Dataset Performance Details

B.1 NSL-KDD ROC-Kurven



(a) Baseline-Modelle (6 Algorithmen)

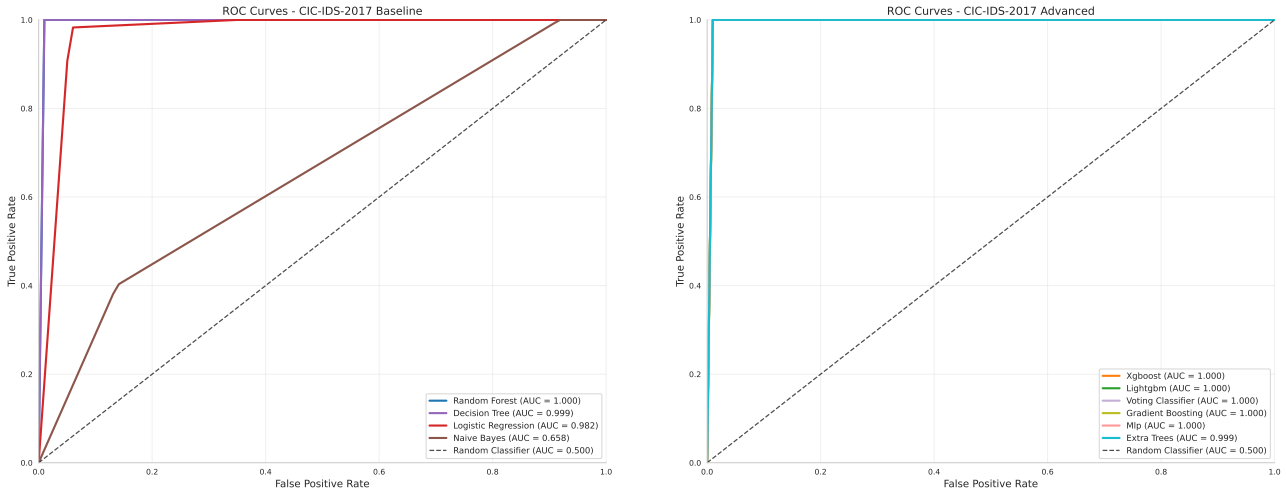
(b) Advanced-Modelle (6 Algorithmen)

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

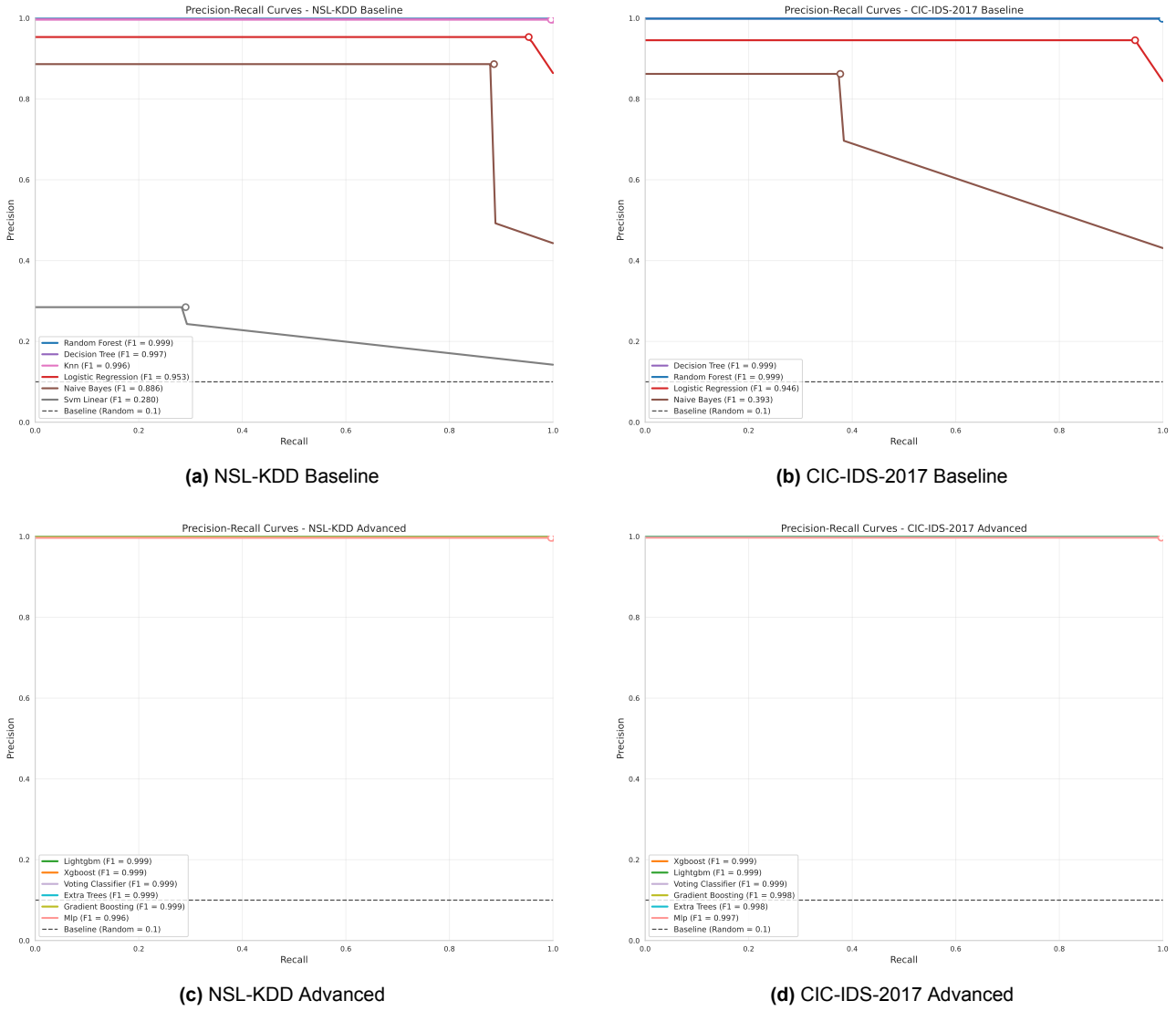


(a) Baseline-Modelle **(b)** Advanced-Modelle

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

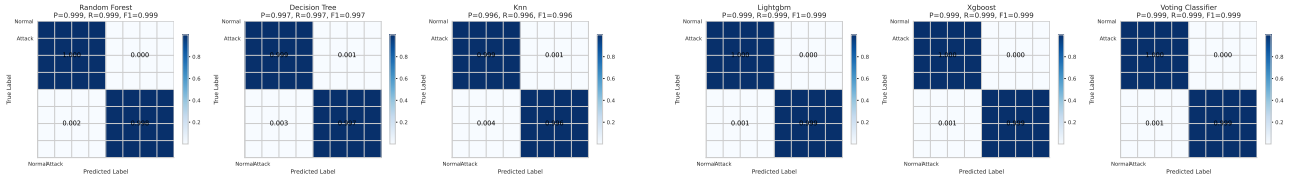
(d) CIC-IDS-2017 Advanced

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

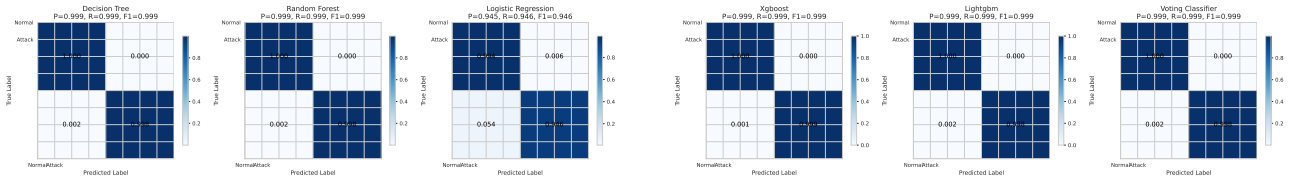


(a) Baseline-Modelle **(b)** Advanced-Modelle

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

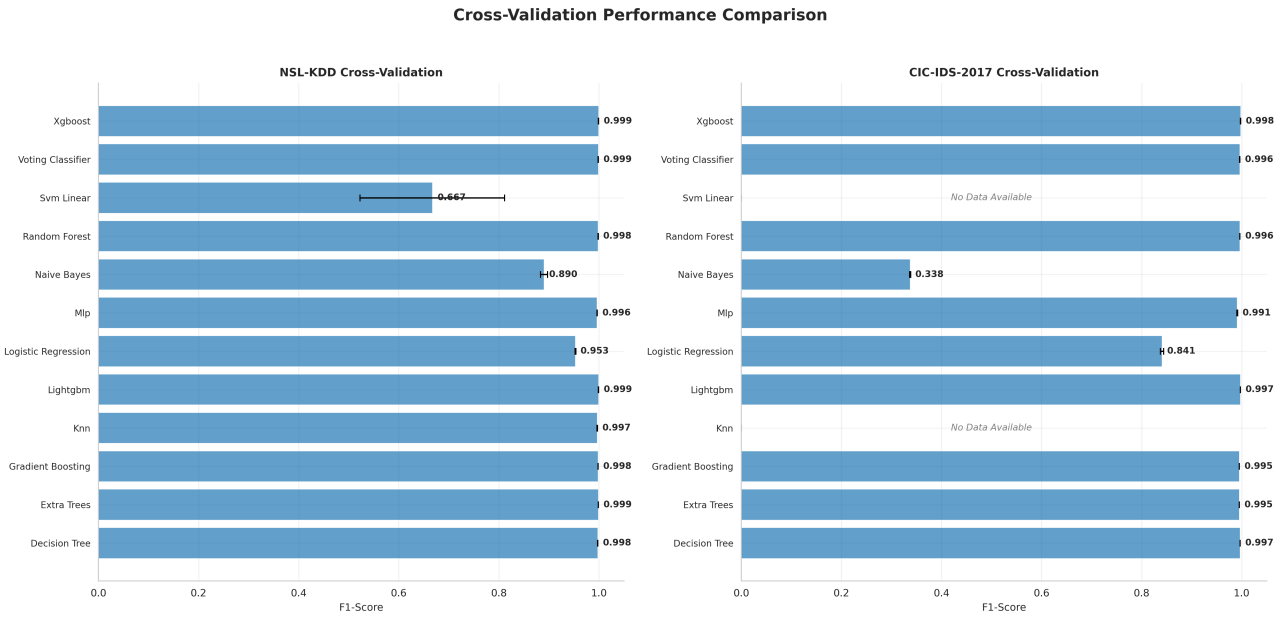


Abb. 12: Cross-Validation Performance-Vergleich NSL-KDD vs. CIC-IDS-2017: 5-Fold stratifizierte CV mit Konfidenzintervallen (95% CI). Fehlerbalken indizieren Variabilität über Folds.

Eigene Darstellung.

C.2 CV Results Distribution

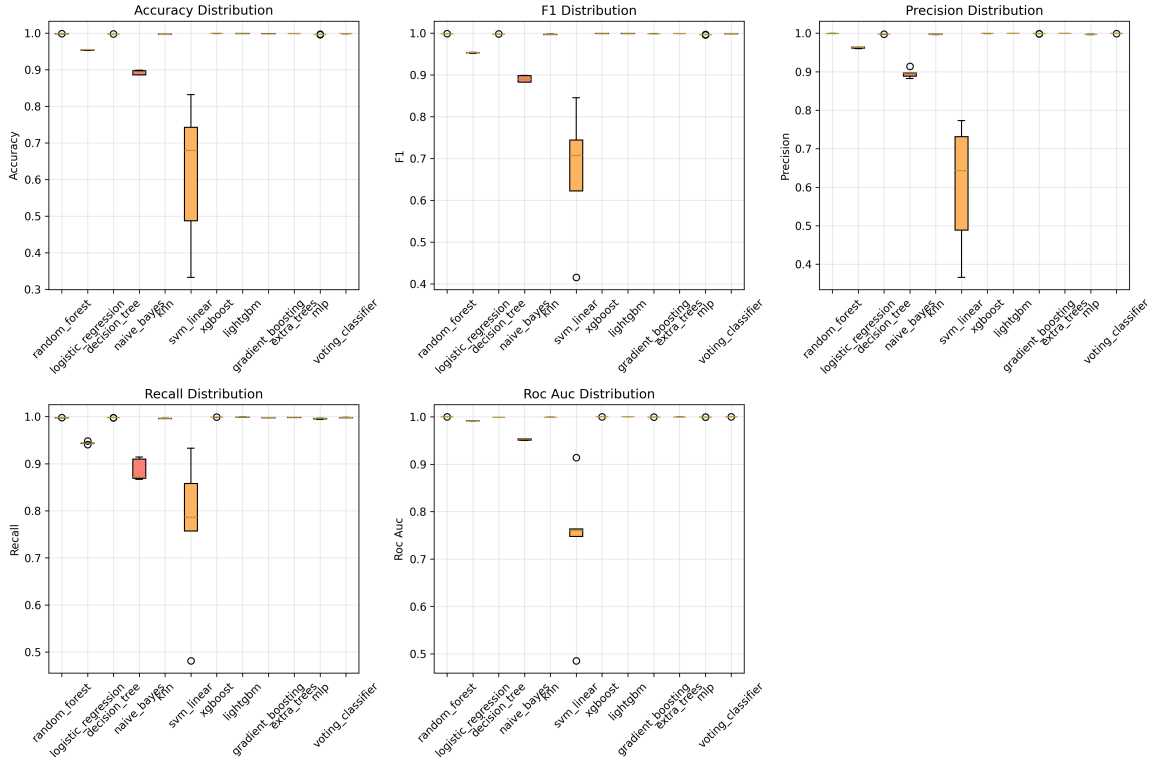


Abb. 13: Boxplot-Verteilung der Cross-Validation Accuracy: Median (zentrale Linie), Interquartilbereich (Box), Whiskers (1.5×IQR), Ausreißer (Punkte). SVM-Linear zeigt extreme Variabilität über Folds (IQR = 0.43, Range = 0.33–0.83).

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

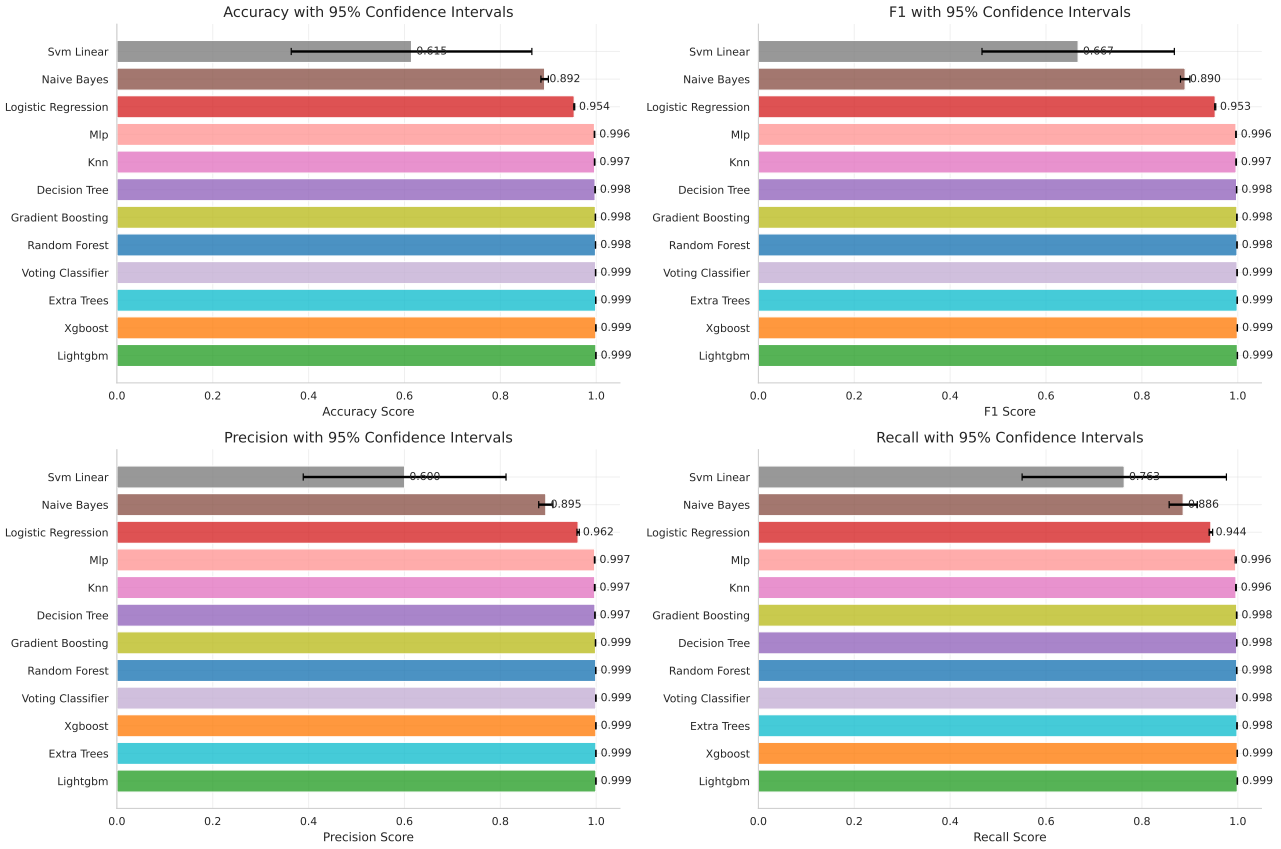


Abb. 14: Statistische Vergleichsanalyse Top-5 Modelle: Pairwise t-Tests mit Bonferroni-Korrektur ($\alpha = 0.01$). Heatmap zeigt p-Werte, Sterne indizieren Signifikanz (*** p < 0.001, ** p < 0.01, * p < 0.05).

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

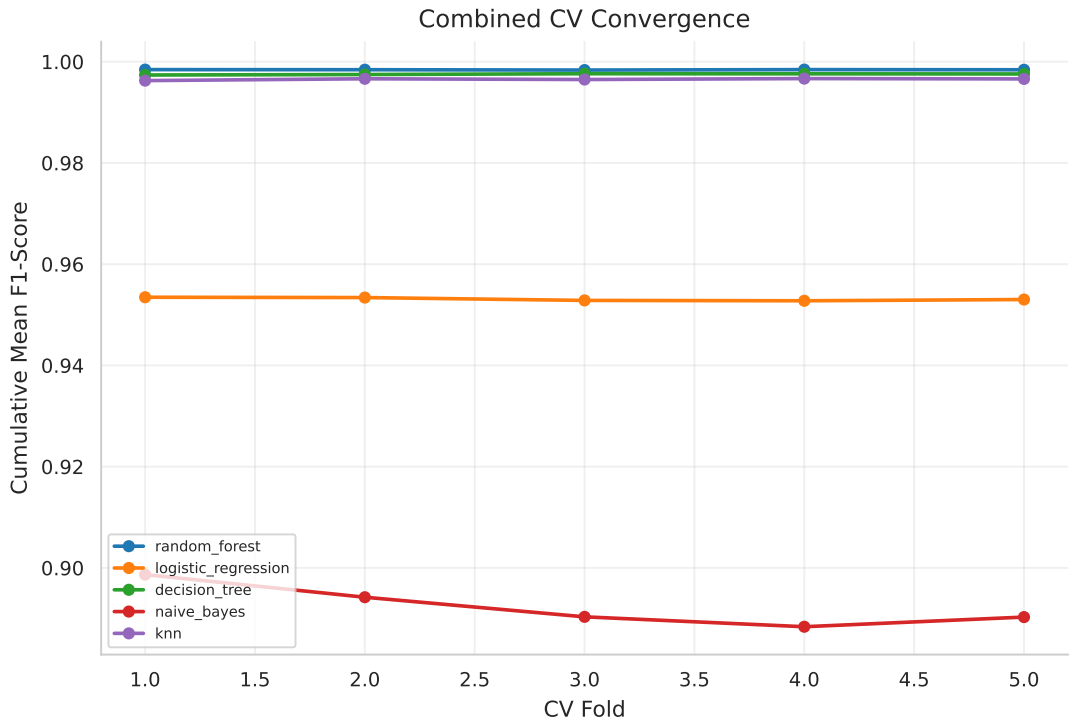


Abb. 15: Cross-Validation Konvergenzanalyse: Kumulative Mean Accuracy ± SD über Folds 1–5. Konvergenz ab Fold 3 indiziert ausreichende k-Wahl. Gestrichelte Linie = finale 5-Fold Mean.

Eigene Darstellung.

Konvergenz-Interpretation

- **Schnelle Konvergenz (Fold 2–3):** XGBoost, LightGBM, Random Forest → stabile Performance
- **Langsame Konvergenz (Fold 4–5):** SVM-Linear, Naive Bayes → 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

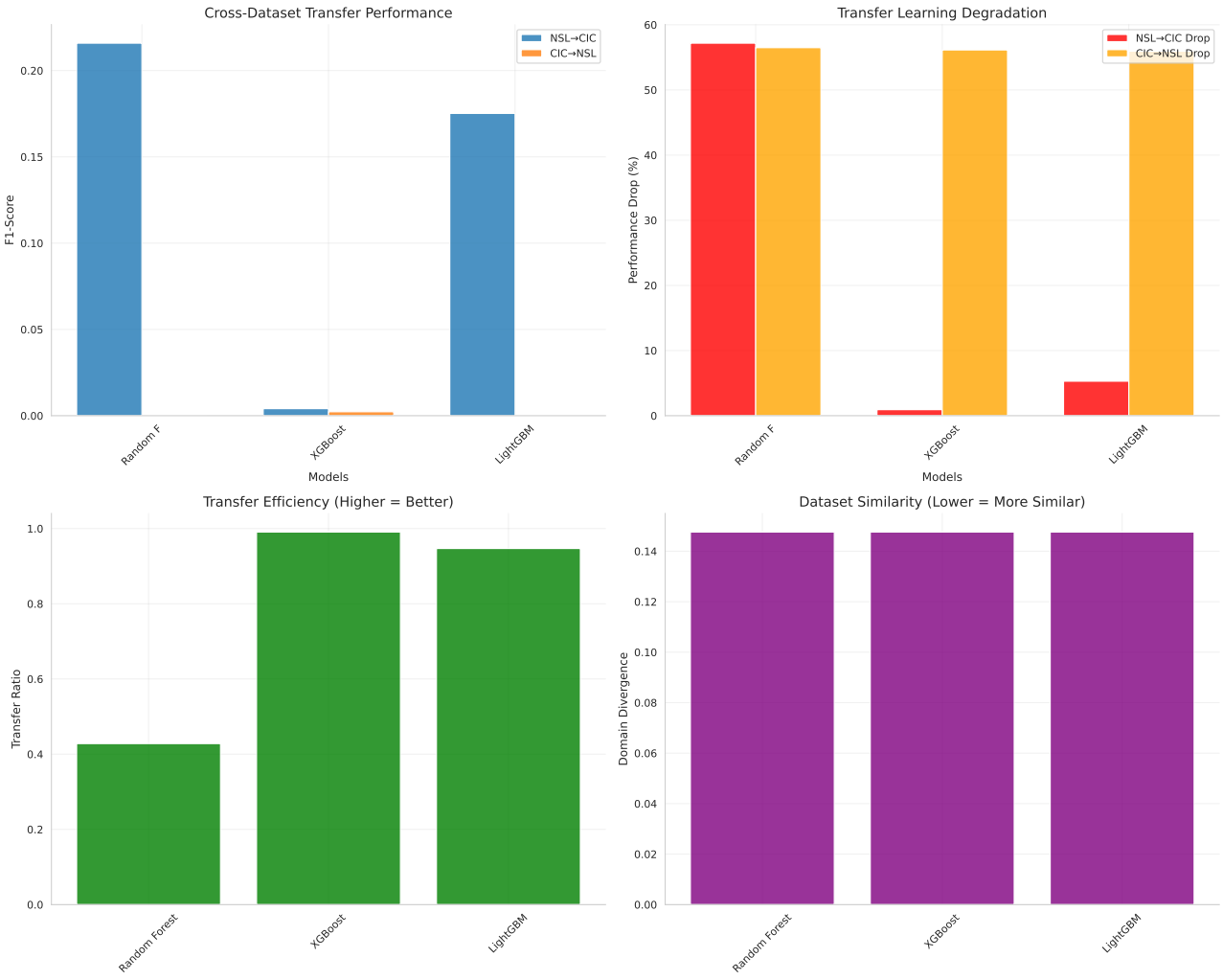


Abb. 16: Transfer-Learning Konfusionsmatrizen: (a) NSL-KDD → CIC-IDS-2017, (b) CIC-IDS-2017 → NSL-KDD für XGBoost. Forward-Transfer (a) zeigt moderate Generalisierung (Target Acc = 0.827), Reverse-Transfer (b) zeigt starke Degradation (Target Acc = 0.431).

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

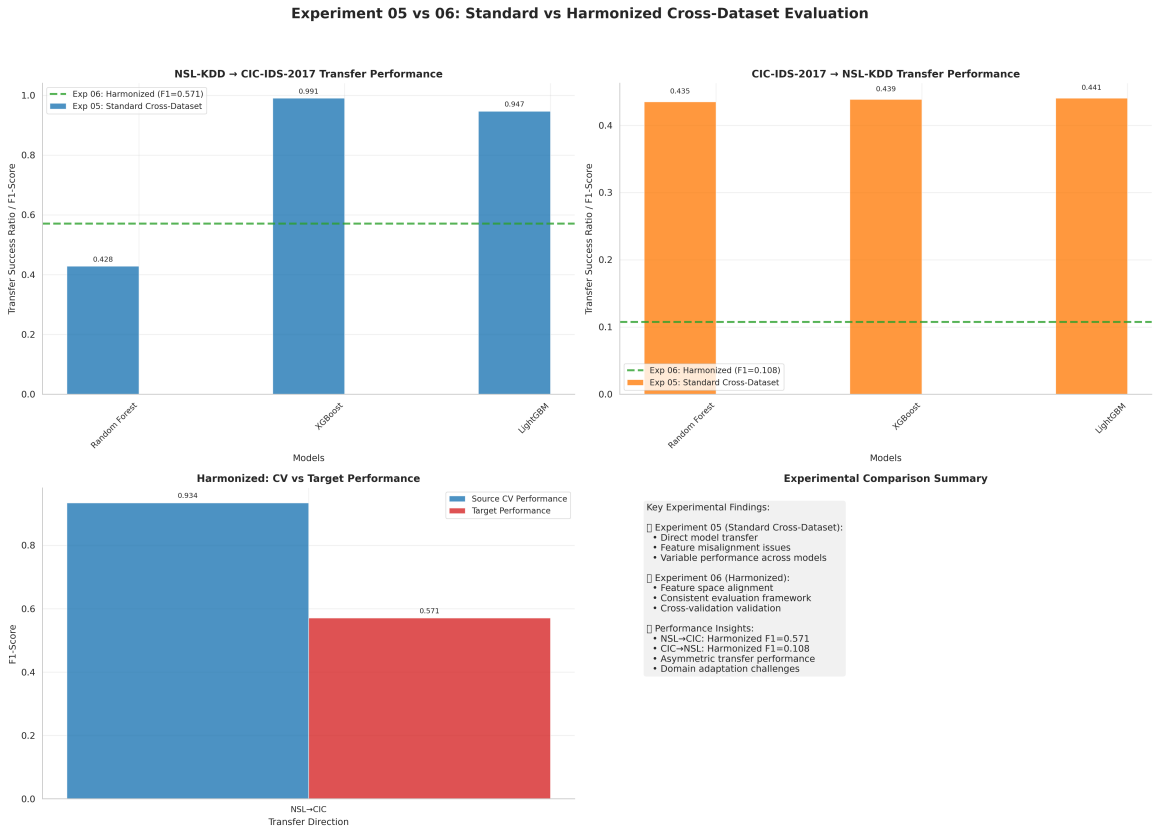


Abb. 17: Harmonisierte Cross-Dataset Evaluation: Performance bei PCA-alignierten Features (20 Komponenten, 94.7% erklärte Varianz). Threshold-Tuning via Grid Search (0.1–0.9 in 0.1-Schritten).

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

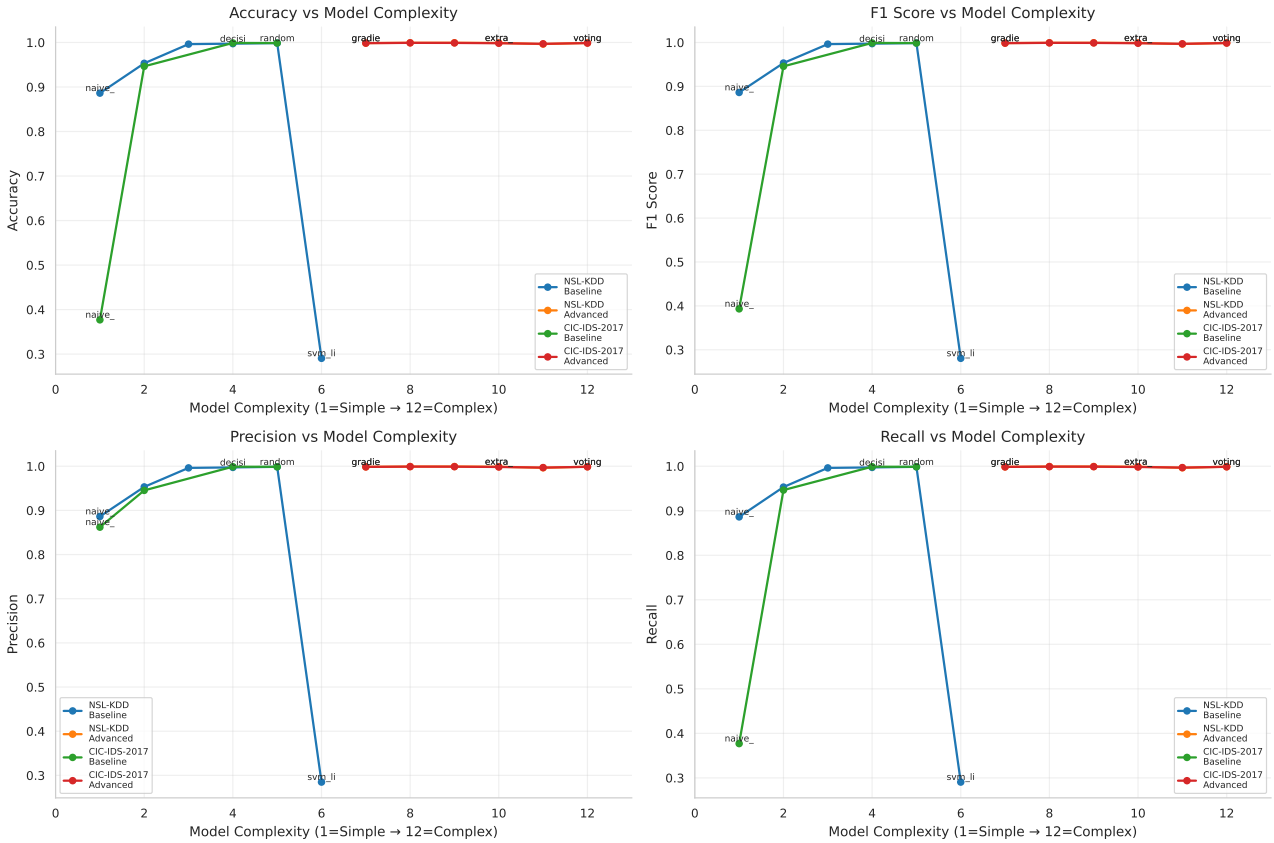


Abb. 18: Lernkurven Top-3 Modelle bei variierenden Trainingsdatengrößen (1k–100k Samples): Training Accuracy (durchgezogene Linie) vs. Validation Accuracy (gestrichelt). Schattierte Bereiche = 95% CI über 3 Wiederholungen.

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

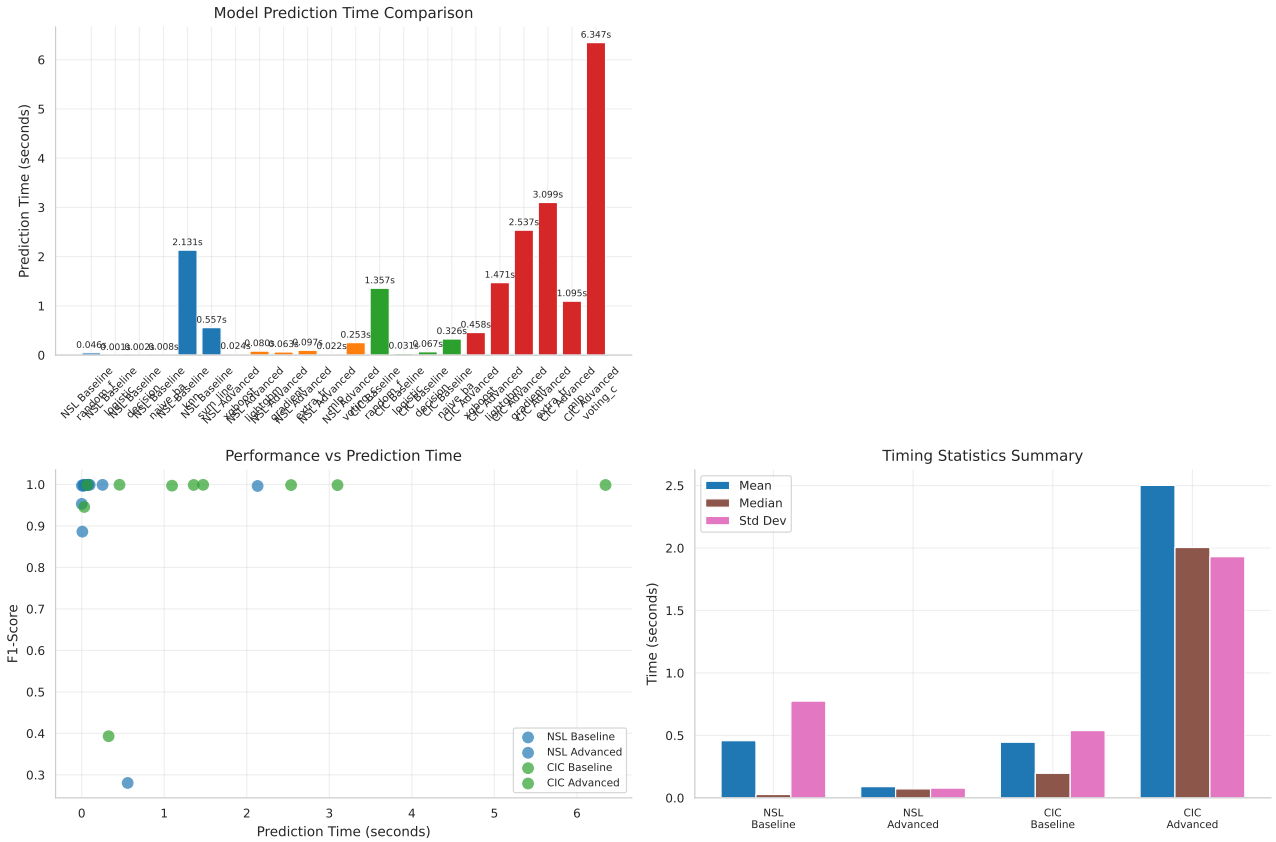


Abb. 19: Training Time vs. Accuracy Trade-Off: Bubble-Chart mit Bubble-Größe proportional zu Inferenzzeit. Optimale Modelle in oberer linker Region (hohe Accuracy, niedrige Training Time).

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. 1: 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

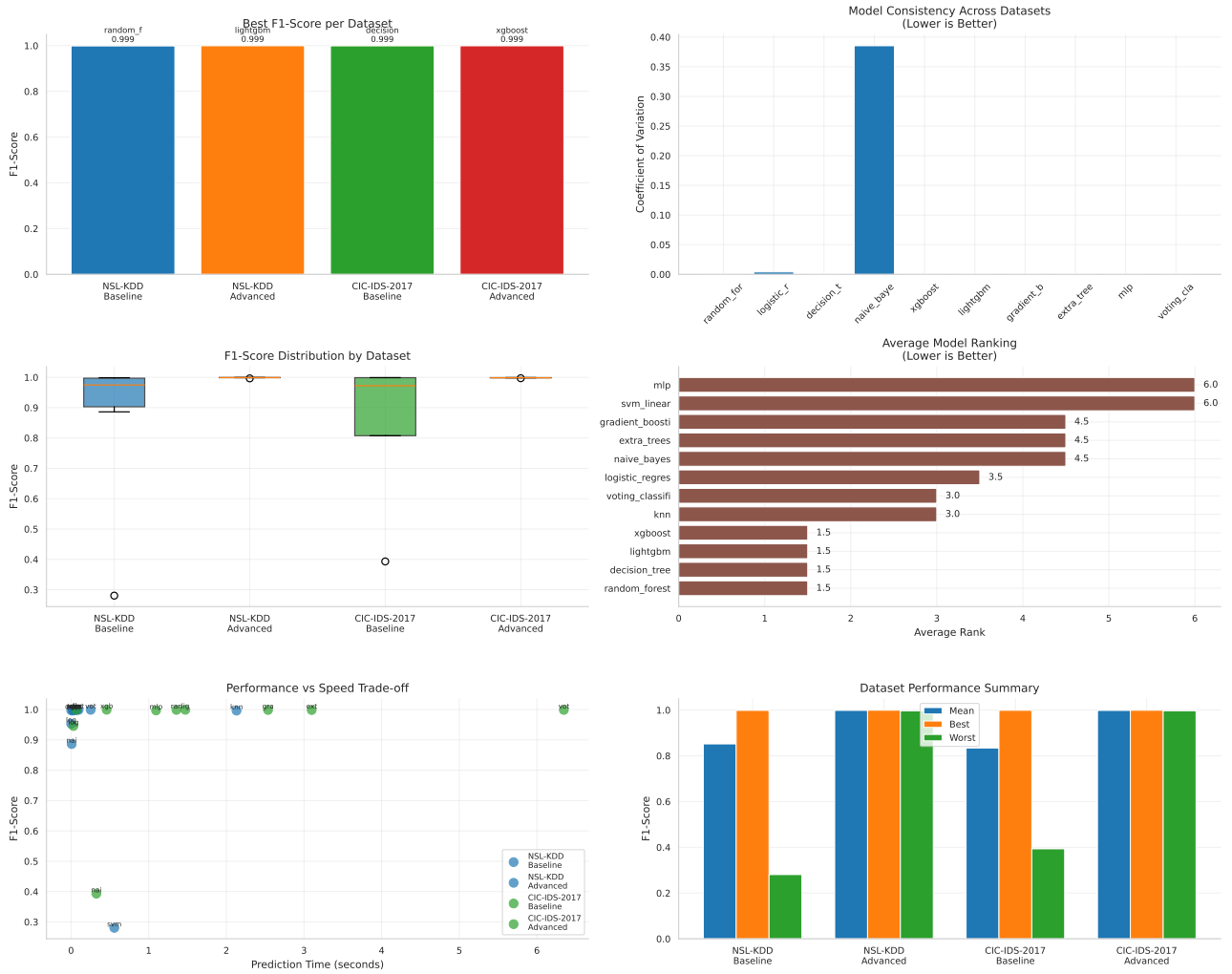


Abb. 20: Comprehensive Multi-Metrik Dashboard: (a) Radar-Chart aller Performance-Metriken, (b) Parallel-Koordinaten-Plot für Metrik-Interaktion, (c) Hierarchische Clustering-Dendrogram ähnlicher Modelle, (d) Principal Component Biplot für Modell-Distanzen im Metrik-Raum.

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)