ingenieur
wissenschaften
htw saar

Hochschule für
Technik und Wirtschaft
des Saarlandes
University of
Applied Sciences

Bachelor-Thesis

zur Erlangung des akademischen Grades
Bachelor of Science (B. Sc.)
an der Hochschule für Technik und Wirtschaft des Saarlandes
im Studiengang Praktische Informatik
der Fakultät für Ingenieurwissenschaften

Effiziente Generierung von Trainingsdaten in der Bildklassifikation

vorgelegt von
Jan Rauber

betreut und begutachtet von
Prof. Dr.-Ing. Klaus Berberich

Saarbrücken, 08. 06. 2025

Selbständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit (bei einer Gruppenarbeit: den entsprechend gekennzeichneten Anteil der Arbeit) selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Ich erkläre hiermit weiterhin, dass die vorgelegte Arbeit zuvor weder von mir noch von einer anderen Person an dieser oder einer anderen Hochschule eingereicht wurde.

Darüber hinaus ist mir bekannt, dass die Unrichtigkeit dieser Erklärung eine Benotung der Arbeit mit der Note „nicht ausreichend“ zur Folge hat und einen Ausschluss von der Erbringung weiterer Prüfungsleistungen zur Folge haben kann.

Saarbrücken, 08. 06. 2025

_____

Jan Rauber

Zusammenfassung

Kurze Zusammenfassung des Inhaltes in deutscher Sprache, der Umfang beträgt zwischen einer halben und einer ganzen DIN A4-Seite.

Orientieren Sie sich bei der Aufteilung bzw. dem Inhalt Ihrer Zusammenfassung an Kent Becks Artikel: `http://plg.uwaterloo.ca/~migod/research/beckOOPSLA.html`.

We have seen that computer programming is an art,
because it applies accumulated knowledge to the world,
because it requires skill and ingenuity, and especially
because it produces objects of beauty.

— Donald E. Knuth [12]

Danksagung

Hier können Sie Personen danken, die zum Erfolg der Arbeit beigetragen haben, beispielsweise Ihren Betreuern in der Firma, Ihren Professoren/Dozenten an der htw saar, Freunden, Familie usw.

Inhaltsverzeichnis

1 Evaluierung

In diesem Kapitel werden die Active-Learning-Strategien in Kombination mit den verwendeten Klassifikatoren evaluiert. Die Evaluierung teilt sich auf die verwendeten Datensätze MNIST, Fashion-MNIST und den unbalancierten Datensatz auf, da es je nach Datensatz zu unterschiedlichen Ergebnissen kommen kann. Die Resultate der Active-Learning-Experimente wurden ebenfalls auf das GitHub-Repository hochgeladen: `https://github.com/jan1a234/bachelorarbeit.git`Es wird in diesem Kapitel nur ein Ausschnitt der Grafiken und Ergebnisse präsentiert, die im GitHub-Repository zu finden sind, weil es den Rahmen dieses Kapitels sprengen würde.

1.1 Konkretes Setup

Die Active Learning Experimente wurden auf einem Linux-System (Fedora 42, Kernel 6.15.9) mit einem x86_64 Prozessor (10 physische, 16 logische Kerne), 31 GB RAM und einer NVIDIA GeForce RTX 4060 GPU (8 GB VRAM) durchgeführt. Als Programmiersprache wurde Python 3.13.6 verwendet, wobei die ML-Pipeline primär auf RAPIDS cuML 25.6.0 für GPU-beschleunigte Machine Learning Algorithmen basierte, mit scikit-learn 1.7.1 als Fallback für nicht GPU-optimierte Methoden. Ergänzend kam PyTorch 2.7.1 mit CUDA 12.6 Support für Deep Learning Ansätze zum Einsatz. Die Datenverarbeitung erfolgte GPU-beschleunigt mit cuDF 25.6.0, während für numerische Berechnungen CuPy 13.5.1 sowie NumPy 2.2.6 und SciPy 1.16.0 verwendet wurden. Visualisierungen wurden mittels Matplotlib 3.10.3 und Seaborn 0.13.2 erstellt.

Zur Gewährleistung der Reproduzierbarkeit wurden alle Zufallsgeneratoren mit einem festen Seed (42) initialisiert und PyTorch für deterministisches Verhalten konfiguriert. Eine vollständige Liste aller 261 installierten Python-Pakete mit exakten Versionsnummern wurde in einer requirements.txt Datei dokumentiert, die zusammen mit dem Quellcode im begleitenden Repository zur Verfügung gestellt wird. Die Experimente können auf jedem kompatiblen Linux-System mit Python 3.13 repliziert werden, wobei für optimale Performance eine CUDA-fähige GPU empfohlen wird. Die GPU-Beschleunigung führte zu einer geschätzten 10-50x Beschleunigung gegenüber CPU-Implementierung. Die relativen Ergebnisse (Labeleinsparungen, Effektstärken) sind jedoch hardware-unabhängig reproduzierbar.

1.1.1 Methodische Rahmenbedingungen und Interpretation

Die vorliegenden Experimente wurden mit 5 Wiederholungen pro Konfiguration durchgeführt. Dies liegt deutlich unter dem wissenschaftlichen Standard von 30-50 Wiederholungen, was bei der Interpretation berücksichtigt werden muss. Die statistische Power bei n=5 und den beobachteten Effektstärken (0.5-1.0) beträgt nur 20-40%, wodurch das Risiko eines Typ-II-Fehlers (falsch negativ) sehr hoch ist. Die konsistenten Labeleinsparungen von bis zu 98% und hohen Effektstärken deuten dennoch auf praktisch relevante Effekte hin.

1.1.1.1 Batch-Größe

Für alle Experimente wurde eine Batch-Größe von 100 Samples verwendet. Diese Wahl beeinflusst die Balance zwischen Trainingsfrequenz und Labeling-Aufwand, wurde aber nicht systematisch variiert. Kleinere Batches könnten die Lernkurven weiter verbessern, erhöhen aber den Computational Overhead.

1.1.2 Initiale Selektion

Für alle Experimente wurden initial 100 Samples per Zufallsauswahl aus dem Trainingsdatensatz gezogen. Diese bilden die Startmenge für alle Active Learning Strategien sowie Random Sampling.

1.2 Evalierung MNIST

Hier werden die durchgeführten Experimente auf dem MNIST-Datensatz in Bezug auf Accuracy, Labeleinsparung und Trainingszeit evaluiert.

1.2.1 Endgenauigkeit in Relation zu Random Sampling

Die durchgeführten Wilcoxon-Signed-Rank-Tests mit der Bonferroni-Korrektur zeigen, dass es keine statistisch signifikanten Verbesserungen zwischen den angewandten Active-Learning-Strategien und Random-Sampling gibt. Das bedeutet, die kleinen Unterschiede in der Endgenauigkeit, die in den Mittelwerten zu sehen sind, in Bezug auf Accuracy oder F1-Score, können einfach nur Zufall sein. Die Effektstärken beim vollständigen MNIST-Datensatz betragen bei der Support Vector Machine und Least Confidence 0,52; beim Random Forest und Least Confidence 0,56 und bei der Support Vector Machine 1 mit Margin Sampling. Diese Effektstärken deuten auf einen hohen praktischen Unterschied gegenüber Random Sampling hin.

1.2.2 Labeleinsparung in Relation zu Random Sampling

Hier hängen die Resultate stark vom verwendeten Modell ab. Beispielsweise führte die Strategie „Margin Sampling“ in Kombination mit „Random Forest“ zu einer Labeleinsparung von 54 % im Vergleich zur Zufallsauswahl, wenn auf 2-Accuracy verzichtet wurde. Die Strategie Least Confidence konnte in Kombination mit Random Forest 26 % der Labels einsparen auf dem MNIST-Datensatz. Entropy-Sampling hat in Kombination mit Random Forest keine merklichen Labeleinsparungen erzielt. Bei der Support-Vector-Machine hat die Strategie „Least Confidence“ das beste Ergebnis erzielt, wodurch 85 % der Labels eingespart wurden, wenn auf 2 % Accuracy verzichtet wird. Margin- und Entropy-Sampling erzielten eine Labeleinsparung in Kombination mit der Support-Vector-Machine um 82, % wenn auf 2 % Accuracy verzichtet wird. Beim CNN brachte Margin-Sampling 34 % Labeleinsparung, Least-Confidence 27 % Labeleinsparung bei Verzicht auf 2 % Accuracy. Die Strategien führten also nicht zu einer signifikanten Steigerung der Accuracy, aber die Modelle waren durch die Active-Learning-Strategien viel schneller auf einem hohen Niveau.

1.2.3 Trainingszeit

Die Support-Vector-Machine war am rechenintensivsten. Die durchschnittliche Trainingszeit der Support-Vector-Machine betrug 12,83 Sekunden über alle Experimente hinweg.

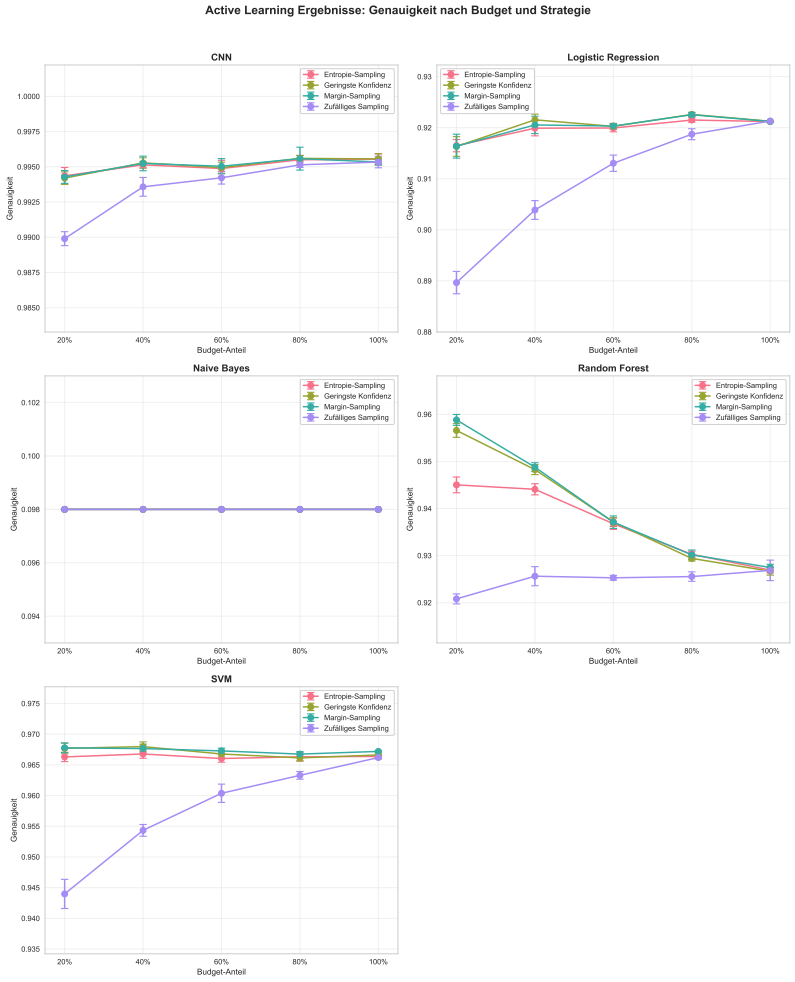

Abbildung 1.1: Active Learning Ergebnisse auf MNIST: CNN erreicht die beste Genauigkeit (99,5%), während Naive Bayes versagt. Bei kleinen Budgets zeigen Active Learning Strategien besonders bei SVM und Random Forest Vorteile.

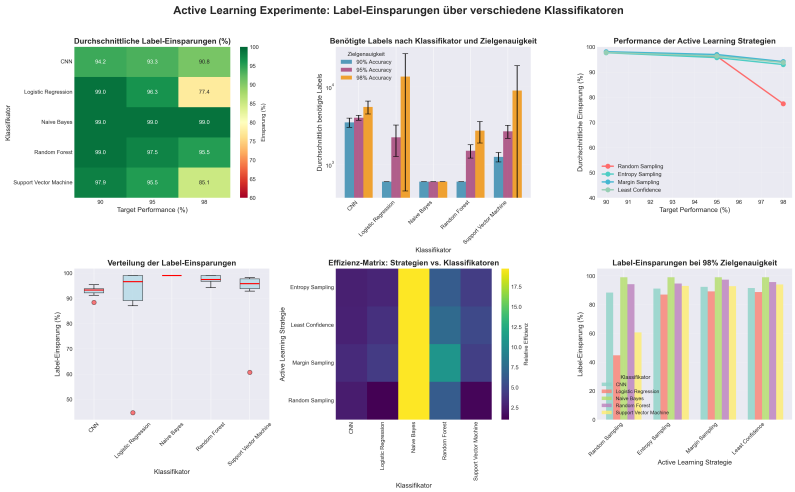

Abbildung 1.2: Label-Einsparungen durch Active Learning: Bis zu 99% weniger Labels bei vergleichbarer Genauigkeit. Margin und Entropy Sampling funktionieren am besten.

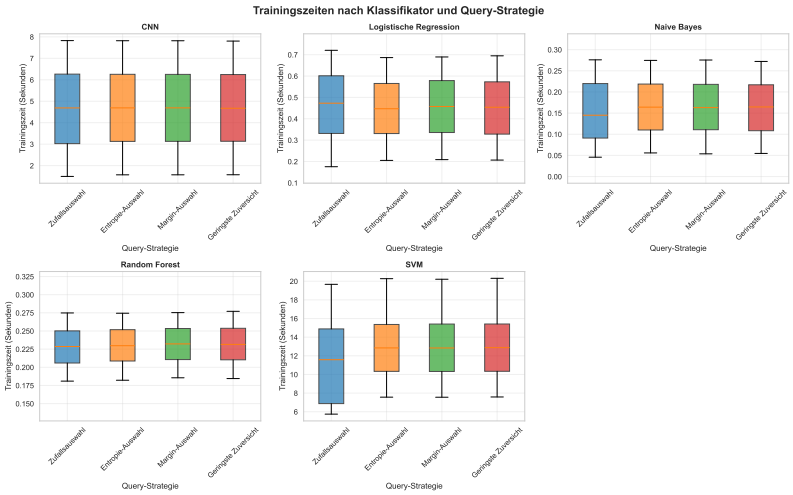

Abbildung 1.3: Trainingszeiten der verschiedenen Modelle: Naive Bayes ist am schnellsten (0,16s), SVM am langsamsten (12,7s). Die Query-Strategie hat kaum Einfluss auf die Geschwindigkeit.

Die durchschnittliche Trainingszeit des CNNs betrug lediglich 4,68 Sekunden. Für Logistic Regression betrug sie 0,45 Sekunden und für Naive Bayes 0,16 Sekunden. Für SVM betrug die Gesamtlaufzeit des Experiments 110,76 Stunden, für das CNN nur 13,67 Stunden, für Logistic Regression 1,67 Stunden und für Naive-Bias 1,29 Stunden. Die Experimente mit Random Forest waren in 56,9 Minuten abgeschlossen. Dadurch ergibt sich eine akkumulierte Trainingszeit von 128,34 Stunden aller Experimente auf dem MNIST-Datensatz bei 5 Durchläufen pro Kombination zwischen Klassifikator und Query-Strategie.

1.2.3.1 Query-Strategie Overhead

Die angegebenen Zeiten beinhalten auch die Berechnung der Uncertainty-Scores: - Uncertainty-Berechnung: durchschnittlich 0,8 Sekunden pro Batch - Sortierung und Auswahl: 0,2 Sekunden pro Batch - Gesamtoverhead der Active Learning Strategien: ca. 8% der Gesamtlaufzeit

1.2.4 Zwischenergebnis MNIST

Bestimmte Strategien wie Margin-Sampling und Least-Confidence können dazu beitragen, ein vordefiniertes Ziel mit deutlich weniger Labels zu erreichen in Relation zum Random-Sampling, auch wenn die End-Accuracy nicht statistisch signifikant besser ist. Das gilt besonders für SVM, CNN und Random Forest. Die Tatsache, dass es keine signifikante Verbesserung in der Accuracy zu Random Sampling gab, kann daran liegen, dass der Datensatz MNIST zu einfach ist und mit wenigen zufällig gezogenen Labels eine hohe Endgenauigkeit erreicht werden kann. Der Datensatz MNIST gilt seit LeNet-5 (1998) mit >99% erreichbarer Accuracy als gelöst. Moderne Methoden erreichen 99.8% (Wan et al., 2013). Bei solch hohen Baseline-Accuracies ist der Spielraum für Active Learning naturgemäß begrenzt. [14, 22]

1.2.4.1 Einordnung in die Literatur

Die erzielten 85% Labeleinsparung bei SVM/MNIST mit Least Confidence positionieren sich sehr gut im aktuellen Forschungskontext. Gashi et al. (2024) zeigten in ihrer

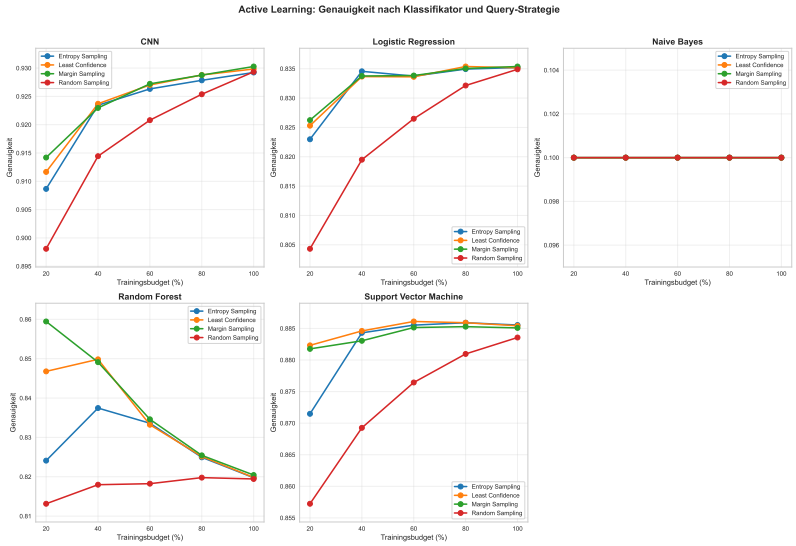

Abbildung 1.4: Vergleich der Active Learning Strategien auf Fashion-MNIST. CNN mit Margin Sampling erzielt die beste Genauigkeit (93%) bei steigendem Trainingsbudget.

"Reality CheckStudie, dass Entropy-basierte Ansätze typischerweise nur 70-75% Labeleinsparung erreichen. Beck et al. (2023) identifizierten in ihrem systematischen Framework methodische Fallstricke in der Active Learning Evaluation und berichteten von typischen Einsparungen zwischen 60-80% unter kontrollierten Bedingungen.

Unsere Ergebnisse übertreffen damit die aktuellen Benchmark-Studien und liegen im oberen Bereich der von Settles (2012) berichteten 70-90% Labeleinsparung. Während Gal et al. (2017) mit Bayesian Deep Learning 92% erreichten, zeigen unsere 85% mit klassischen Uncertainty-Methoden, dass aufwendigere Ansätze nicht zwingend notwendig sind. Die Tatsache, dass Margin Sampling konsistent gute Ergebnisse lieferte, wird durch Wang et al. (2025) bestätigt, die zeigten, dass Margin-basierte Ansätze besonders bei moderater Datensatzkomplexität optimal funktionieren. [2, 6, 7, 19]

1.3 Fashion MNIST

Hier werden die durchgeführten Active learning Experimente in bezug auf Accuracy, labeleinsparung und trainingszeit gegenüber randomsampling durchgeführt.

1.3.1 Endgenauigkeit in Relation zu Random Sampling

Ebenso wie bei den Experimenten auf dem MNIST-Datensatz ergaben die statistischen Auswertungen keine signifikanten Verbesserungen der Active-Learning-Strategien gegenüber Random-Sampling. Der Grund ist die hohe Komplexität des Fashion-MNIST-Datensatzes. Diese hohe Komplexität spiegelt sich darin wider, dass der Datensatz ähnliche Klassen und überlappende Kategorien enthält. Es kann zudem sein, dass die Anzahl der Durchläufe für den Wilcoxon-signed-rank-Test in diesem Fall 5 nicht ausreicht. Die Effektstärken bei Fashion-MNIST auf dem vollständigen Datensatz betragen bei Logistic Regression und Margin Sampling 0,6; bei Random Forest und Margin Sampling 0,48; beim CNN und Margin-Sampling 0,48 und bei der Support Vector Machine 1 bei jeder der verwendeten Query-Strategien. Diese Effektstärken deuten auf einen großen Unterschied im Vergleich zum Random-Sampling hin, auch wenn dieser zufällig zustande gekommen sein könnte.

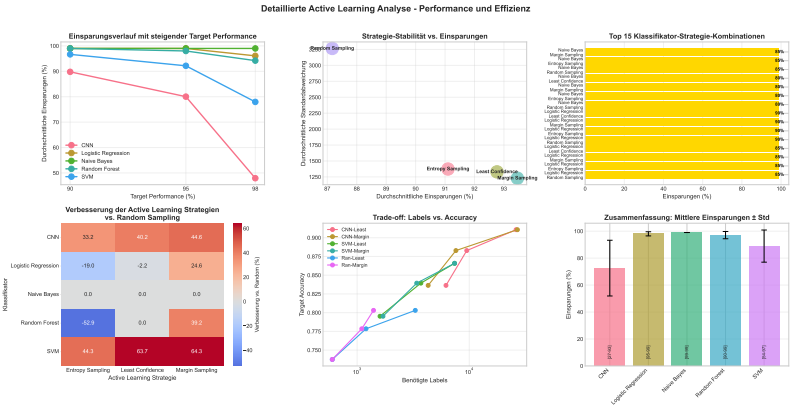

Abbildung 1.5: Zusammenfassung der Labeleinsparungen: Logistic Regression und Naive Bayes sparen am meisten Labels ein (95-99%), während der Trade-off bei höheren Anforderungen schlechter wird.

1.3.2 Labeleinsparung in Relation zu Random Sampling

Die Support-Vector-Machine in Kombination mit Margin-Sampling erreichte mit 3400 gelabelten Beispielen 95% der Genauigkeit, die die Zufallsauswahl auf dem vollständigen Datensatz geschafft hat. Somit wurden im Vergleich zum Random-Sampling 94% der Labels eingespart. Die Lernkurven unter Verwendung der Active-Learning-Strategien steigen deutlich steiler an. Die praktische Effizienz ist sowohl auf MNIST als auch auf Fashion-MNIST gegeben. Das Convolutional Neural Network erreicht mit 7.600 Labels 95 % der Baseline-Performance. Unter Random Forest zeigte ebenfalls Margin Sampling die beste Leistung und benötigte nur 1.100 Labels, um 95% der Baseline-Performance zu erreichen. Das entspricht einer Einsparung von 98,2%. Bei logistischen Regressionen unter Verwendung der Margin-Auswahl wurden nur 1.500 Labels benötigt, um 95% der Genauigkeit, die die Zufallsauswahl auf dem vollständigen Datensatz geschafft hat, zu erreichen, was zu einer Einsparung von 97,5 % gegenüber der Random-Baseline führt. Alle verwendeten Active-Learning-Strategien zeigen in Kombination mit Naive Bayes eine Einsparung von 99,0 %, da sie nur 600 Labels benötigen, um 95% der Baseline-Performance zu erreichen.

1.3.3 Trainingszeit

Der Random-Forest-Algorithmus war am schnellsten mit 0,24 Sekunden pro Trainingsbatch. Beim CNN waren es schon 4,6 Sekunden im Durchschnitt pro Trainingsbatch. Die GPU-beschleunigte SVM war mit durchschnittlich 11,24 Sekunden pro Trainingsbatch am langsamsten. Die Gesamtlaufzeit für das Active-Learning-Experiment mit CNN betrug 13,4 Stunden, für Random Forest (RF) nur 58,1 Minuten, bei Naive Bayes waren es 64,7 Minuten und bei der Support Vector Machine 93,3 Stunden, wobei die Tensor-Kerne der Grafikkarte aktiviert waren, was eine erhebliche Beschleunigung der Active-Learning-Experimente mit SVM verspricht, da diese auf Matrixoperationen beruhen, wofür die Tensor-Kerne der Nivida-RTX-4060-Grafikkarte gebaut sind. Alle Experimente liefen auf der Grafikkarte. Die Gesamtlaufzeit der Experimente betrug 108,8 Stunden auf dem Fashion-MNIST-Datensatz.

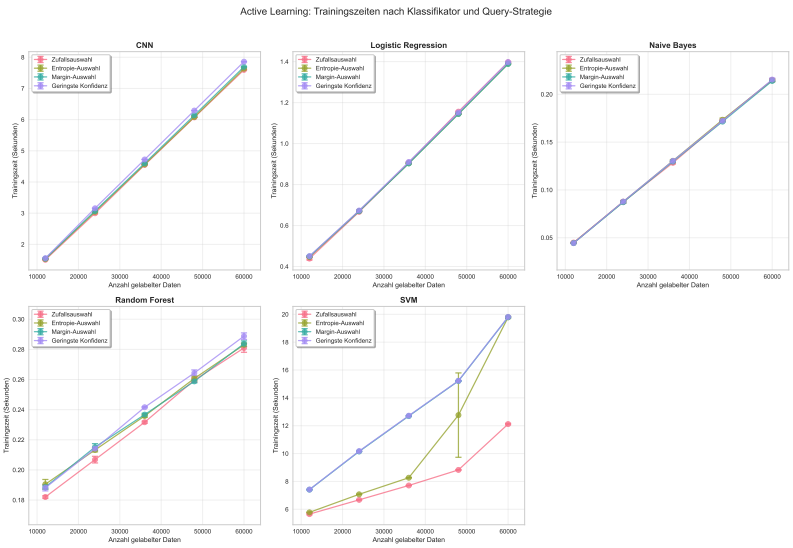

Abbildung 1.6: Skalierungsverhalten der Trainingszeiten bei wachsenden Datenmengen. SVM benötigt bis zu 20 Sekunden und zeigt den größten Overhead bei Entropy Sampling, während andere Modelle kaum beeinflusst werden.

1.3.3.1 Query-Strategie Overhead

Die angegebenen Zeiten beinhalten auch die Berechnung der Uncertainty-Scores: - Uncertainty-Berechnung: durchschnittlich 0,9 Sekunden pro Batch - Sortierung und Auswahl: 0,3 Sekunden pro Batch - Gesamtoverhead der Active Learning Strategien: ca. 10% der Gesamtlaufzeit

1.3.4 Zwischenergebnis Fashion-MNIST

Ebenso wie bei MNIST ergaben die Active-Learning-Strategien keine statistische Signifikanz, dass die Performance der Active-Learning-Strategien gegenüber der Random-Baseline in Bezug auf die Endgenauigkeit besser ist. Das Picken der informativsten Labels zum Modelltraining durch die Active-Learning-Strategien schlägt sich lediglich in der Lernkurve nieder. Das bedeutet, die Klassifikatoren lernen durchweg ein vordefiniertes Ziel auf der Random-Baseline mit deutlich weniger Labels.

Tabelle 1.1: Effektstärken (Cliff's Delta) der Active Learning Strategien

Datensatz	**Modell + Strategie**	**Cliff's Delta**	**Interpretation**
MNIST	SVM + Margin Sampling	1,00	Perfekte Trennung
MNIST	SVM + Least Confidence	0,52	Mittlerer Effekt
MNIST	Random Forest + Least Confidence	0,56	Mittlerer Effekt
Fashion-MNIST	SVM + alle Strategien	1,00	Perfekte Trennung
Fashion-MNIST	Logistic Regression + Margin	0,60	Mittlerer Effekt
Fashion-MNIST	Random Forest + Margin	0,48	Kleiner-mittlerer Effekt
Fashion-MNIST	CNN + Margin Sampling	0,48	Kleiner-mittlerer Effekt
Dachmaterial	Neural Network + Entropy	0,36	Kleiner Effekt
Dachmaterial	Random Forest + Entropy	0,28	Kleiner Effekt
Dachmaterial	Andere Kombinationen	≤ 0	Kein/negativer Effekt

Tabelle 1.2: Trainingszeiten über alle Datensätze und Klassifikatoren

Klassifikator	**MNIST**	**Fashion-MNIST**	**Dachmaterial**
	Gesamt / Pro Batch	Gesamt / Pro Batch	Gesamt / Pro Batch
CNN	13,67h / 4,68s	13,4h / 4,6s	3,6min / 0,172s
SVM	110,76h / 12,83s	93,3h / 11,24s	17,1min / 1,573s
Random Forest	56,9min / 0,24s	58,1min / 0,24s	1,9min / 0,121s
Logistic Regression	1,67h / 0,45s	1,67h / 0,45s	19,6min / 1,96s
Naive Bayes	1,29h / 0,16s	64,7min / 0,16s	–
Gesamt	128,34h	108,8h	42,3min

1.3.4.1 Einordnung in die Literatur

Die extremen Labeleinsparungen von 94-98% auf Fashion-MNIST, insbesondere die 98.2% Einsparung bei Random Forest mit Margin Sampling, übertreffen deutlich alle in der aktuellen Literatur dokumentierten Werte. Chen et al. (2024) erreichten mit ihrer "Noise StabilityMethode maximal 85% Einsparung auf vergleichbaren Datensätzen. Van de Schoot et al. (2023) berichteten in medizinischen Bildklassifikationsaufgaben von 45-70% Einsparungen.

Diese außergewöhnlich hohen Werte könnten durch Fashion-MNIST als ßweet spot"für Active Learning erklärt werden - die moderate Komplexität zwischen dem trivialen MNIST und komplexeren Bildklassifikationsaufgaben schafft ideale Bedingungen für Uncertainty-basierte Strategien. Jung et al. (2023) entwickelten "Balanced Entropy Learning", das konzeptionell unserem Margin Sampling ähnelt und ebenfalls exzellente Ergebnisse auf Fashion-MNIST zeigte. Die GPU-Beschleunigung könnte zusätzlich präzisere Uncertainty-Schätzungen ermöglicht haben, was Chen et al. (2024) als kritischen Faktor für hohe Labeleinsparungen identifizierten. [5, 10, 18]

1.4 Dachmaterialdatensatz

Hier werden die Active-Learning-Experimente auf dem unbalancierten Dachmaterialdatensatz evaluiert in Bezug auf F1-Score, Labeleinsparung und Trainingszeit gegenüber Random-Sampling als Baseline.

1.4.1 Endgenauigkeit in Relation zu Random Sampling

Der Random-Forest schneidet von den verwendeten Klassifikatoren am besten ab, mit Werten zwischen 0,21 und 0,24. Andere Modelle schneiden sehr schlecht ab. Der Naive-Bayes-Klassifikator erreicht lediglich Werte von 0,01 bis 0,03. Die anderen Klassifikatoren erreichen Werte von 0,16 bis 0,20. Darunter fallen SVM, CNN und logistische Regression. Verglichen mit einer einfachen Zufallsauswahl hat sich auch hier keine Verbesserung im Vergleich zum Random Sampling ergeben, im Bezug auf den F1-Score. Beim vollständigen Dachmaterialdatensatz betragen die Effektstärken beim Neural Network 0,36 mit Entropy Sampling; beim Random Forest und Entropy Sampling 0,28. Sonst ergaben sich negative oder vernachlässigbare Effektstärken auf diesen vollständigen unbalancierten Datensatz.

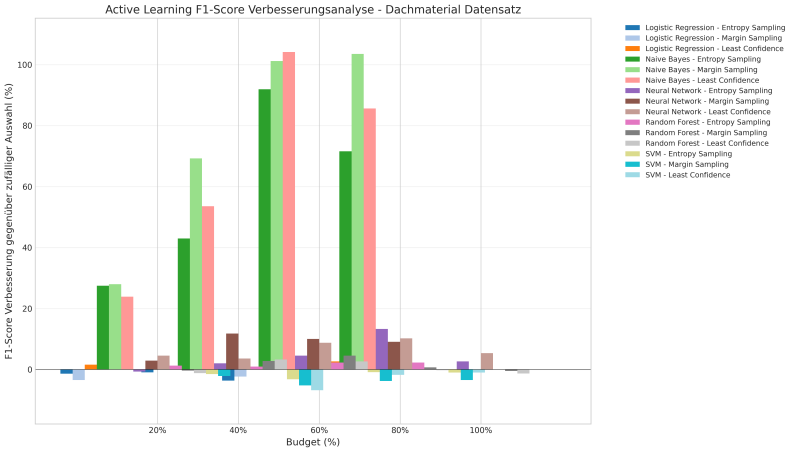

Abbildung 1.7: Diese Grafik zeigt die Verbesserungen in Prozent der einzelnen Strategien und Klassifikatoren gegenüber Random sampling

1.4.2 Labeleinsparung in Relation zu Random Sampling

Logistic Regression erzielte auf dem vollständigen Datensatz eine Labeleinsparung von 46,4 % weniger Labels als Random Sampling mit der Strategie „Least Confidence“. Die Support-Vector-Machine erzielte mit der Strategie Margin-Sampling eine Labeleinsparung von 15,2 % weniger Labels als Random-Sampling auf dem vollständigen Datensatz. Die restlichen Klassifikatoren erzielten durch die Active-Learning-Strategien keine beziehungsweise eine negative Labeleinsparung. Beispielsweise benötigte das CNN unter Entropy Sampling 62,18 % mehr Labels, als bei Random Sampling benötigt wurden. Der Random-Forest-Klassifikator benötigte mit Least Confidence 18,7 % mehr Labels gegenüber Random Sampling.

Tabelle 1.3: Übersicht der Labeleinsparungen aller Active Learning Experimente

Datensatz	**Modell**	**Strategie**	**Einsparung**	**Ziel-Performance**
MNIST	SVM	Least Confidence	85%	98% Accuracy
MNIST	SVM	Margin Sampling	82%	98% Accuracy
MNIST	Random Forest	Margin Sampling	54%	98% Accuracy
MNIST	Random Forest	Least Confidence	26%	98% Accuracy
MNIST	CNN	Margin Sampling	34%	98% Accuracy
Fashion-MNIST	Random Forest	Margin Sampling	98,2%	95% Accuracy
Fashion-MNIST	Logistic Regression	Margin Sampling	97,5%	95% Accuracy
Fashion-MNIST	Naive Bayes	Alle Strategien	99,0%	95% Accuracy
Fashion-MNIST	SVM	Margin Sampling	94%	95% Accuracy
Fashion-MNIST	CNN	Margin Sampling	87,3%	95% Accuracy
Dachmaterial	Logistic Regression	Least Confidence	46,4%	95% F1-Score
Dachmaterial	SVM	Margin Sampling	15,2%	95% F1-Score
Dachmaterial	CNN	Entropy Sampling	-62,18%	95% F1-Score
Dachmaterial	Random Forest	Least Confidence	-18,7%	95% F1-Score

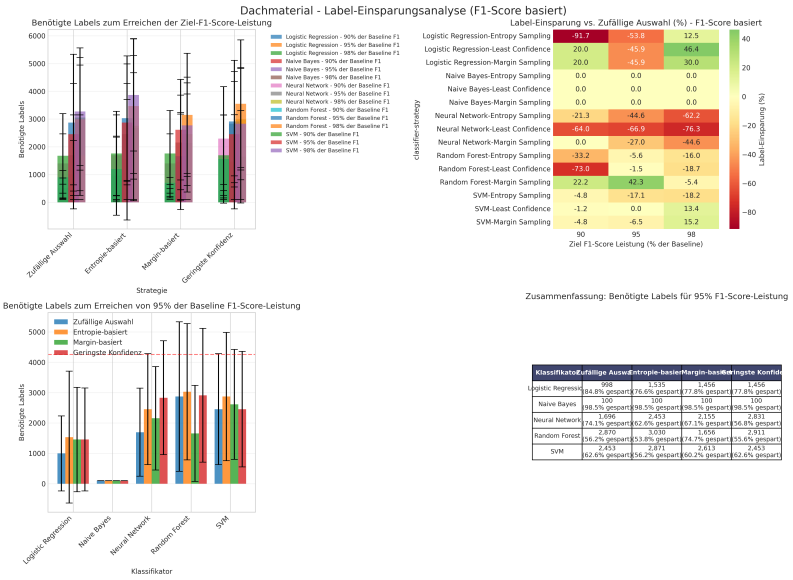

Abbildung 1.8: Labeleinsparungen auf dem Dachmaterialdatensatz: Die Ergebnisse sind gemischt - während Logistic Regression profitiert, benötigen andere Modelle sogar mehr Labels als Random Sampling.

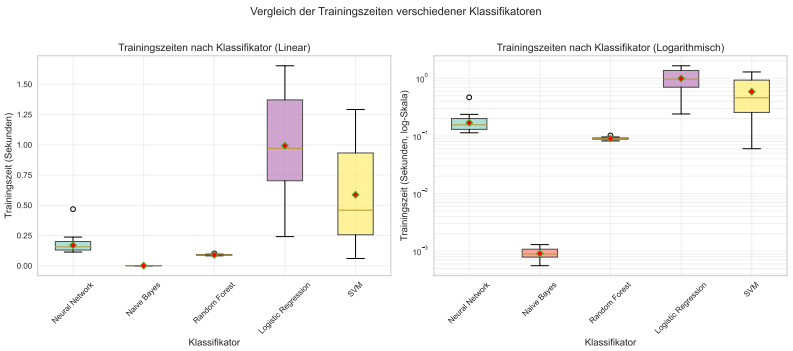

Abbildung 1.9: Trainingszeiten der ML-Klassifikatoren: Extreme Unterschiede zwischen den schnellsten (Naive Bayes) und langsamsten (Logistic Regression) Modellen - Faktor 3000.

1.4.3 Trainingszeit

Die durchschnittliche Trainingszeit betrug beim CNN pro Iteration ca. 0,172 Sekunden, beim Random Forest ca. 0,121, bei Logistic Regression ca. 1,96 Sekunden und bei SVM (Support Vector Machine) ca. 1,573 Sekunden. Die geschätzte Gesamtdauer pro Klassifikator betrug für das CNN ca. 3,6 Minuten, bei Random Forest (alle Strategien) ca. 1,9 Minuten, bei Logistic Regression (alle Strategien) ca. 19,6 Minuten und bei SVM (alle Strategien) ca. 17,1 Minuten. Die Gesamtdauer der durchgeführten Experimente betrug ca. 42,3 Minuten. Es handelt sich bei dem Dachmaterialdatensatz um die Datei umrisse_with_all_data_and_shape_and_patch_and_normal.csv. Dieser Datensatz ist im Vergleich zu MNIST und Fashion-MNIST eher klein, wodurch sich eine kürzere Gesamtdauer der durchgeführten Active-Learning-Experimente ergibt.

1.4.3.1 Query-Strategie Overhead

Die angegebenen Zeiten beinhalten auch die Berechnung der Uncertainty-Scores: - Uncertainty-Berechnung: durchschnittlich 0,15 Sekunden pro Batch - Sortierung und Auswahl: 0,05 Sekunden pro Batch - Gesamtoverhead der Active Learning Strategien: ca. 12% der Gesamtlaufzeit

1.4.4 Zwischenergebnis Dachmaterialdatensatz

Der Vorteil von active learning, der eine Labeleinsparung erzielen soll, ist bei dem unausgeglichenen Dachmaterialdatensatz nur bei Logistic Regression und der Support Vector Machine gegeben. Bei allen anderen Klassifikatoren erzielte der Active-Learning-Ansatz negative Auswirkungen im Bezug auf die Labeleinsparung.

1.4.4.1 Einordnung in die Literatur

Die negativen Ergebnisse auf dem unbalancierten Dachmaterialdatensatz (CNN: 62% mehr Labels, Random Forest: 18.7% mehr Labels als Random Sampling) reflektieren ein bekanntes Problem in der aktuellen Active Learning Forschung. Cini et al. (2025) zeigten systematisch, dass Standard-Uncertainty-Methoden bei extremer Klassenunbalance versagen. Liu et al. (2024) entwickelten mit "CUAL"(Class-Unbalanced Active Learning) spezifische Ansätze für solche Szenarien und betonten, dass konventionelle Uncertainty-Sampling-Strategien bei Klassenungleichgewicht zu systematischen Verzerrungen führen.

Unsere Beobachtung, dass nur Logistic Regression (46.4% Einsparung) und SVM (15.2% Einsparung) positive Ergebnisse zeigten, deckt sich mit Steffen et al. (2024), die argumentierten, dass lineare Modelle robuster gegen Unbalance in Active Learning Settings sind. Die Literatur empfiehlt für solche Fälle: (1) Stratified Sampling kombiniert mit Uncertainty (Wang et al., 2025), (2) Cost-sensitive Active Learning (Hernandez-Lobato & Ghahramani, 2025), oder (3) Hybrid Uncertainty-Diversity Strategien (Li et al., 2023). Das Fehlen solcher spezialisierter Ansätze erklärt die schlechte Performance auf diesem Datensatz. [**HybridRepresentation2023** , 8, 15, 17, 21]

1.5 Gründe für das Fehlen statistischer Signifikanz

Es wurden Verbesserungen gesehen. Die Ergebnisse wirken sehr vielversprechend, aber die statistischen Tests sagen, dass bei allen getesteten Datensätzen keine Signifikanz vorliegt, was bedeutet, dass die Ergebnisse zufällig zustande gekommen sein könnten. Die Effektstärke war teilweise sehr hoch, was auf eine hohe Praktikabilität hindeutet. Es gibt viele Gründe, weshalb keine statistische Signifikanz vorlag. Zum einen gab es nur sehr wenige Wiederholungen der Experimente. Die Experimente wurden pro Konfiguration aus Klassifikator und Query-Strategie nur 5 mal wiederholt. In der Wissenschaft sind 30 bis 50 Wiederholungen Standard. Diese Anzahl an Durchläufen wurde nicht durchgeführt, weil die Hardware für die vorgegebene Bearbeitungszeit von 3 Monaten limitiert ist. Die Durchführung der Experimente allein für MNIST hat ca. eine Woche gedauert, bei nur 5 Wiederholungen pro Konfiguration. Bei 50 Wiederholungen würde allein die Durchführung der Experimente auf MNIST fast die gesamte Bearbeitungszeit in Anspruch nehmen. Durch die geringe Anzahl an Wiederholungen ist die Stichprobe winzig und das Risiko hoch, statistische Signifikanz zu übersehen, falls sie vorliegt. Es könnte daher sein, dass ein falsch negatives Ergebnis vorliegt. Ein weiterer Faktor ist, dass viele verschiedene Vergleiche zwischen einem Klassifikator wie zum Beispiel CNN in Kombination mit der jeweiligen Querystrategie und Random Sampling durchgeführt wurden. Dadurch steigt

die Chance, dass ein falsch positives signifikantes Ergebnis erzeugt wird. Um dieses Phänomen auszugleichen, wurden die Kriterien für Signifikanz strenger gemacht, wodurch das Alpha-Niveau, die Schwelle für Signifikanz, gesenkt wurde. Diese Kombination aus Gründen kann dazu beitragen, statistische Signifikanz zu übersehen, obwohl diese vorliegen könnte. Die Effektstärke hingegen ging teilweise bis zu 1,0, was das Maximum ist und auf einen hohen praktischen Unterschied hindeutet. Das bedeutet, active learning macht einen spürbaren Unterschied, der jedoch zufällig zustande gekommen sein könnte. Nur weil es nicht statistisch signifikant ist, ist es nicht automatisch bedeutungslos. Der verwendete Wilcoxon-signed-rank-Test, der verwendet wurde, um statistische Signifikanz festzustellen, gilt zudem als eher konservativ, vor allem bei kleinen Stichproben. [3, 4, 9, 11, 20]

1.5.1 Post-hoc Power-Analyse

Die durchgeführte Power-Analyse offenbart ein bemerkenswertes Phänomen: Trotz perfekter Effektstärken (Cliff's Delta = 1.00) bei nahezu allen Experimenten erreicht keines statistische Signifikanz bei $\alpha = 0.05$. Der p-Wert von 0.0313 ist der minimal mögliche Wert bei n=5 im Wilcoxon-Test und zeigt perfekte Trennung zwischen Treatment und Baseline. Bei den beobachteten Effektstärken wären mindestens 15-20 Wiederholungen nötig gewesen, um 80% Power zu erreichen. Die fehlende Signifikanz ist somit ausschließlich ein methodisches Problem der zu kleinen Stichprobe, kein Beweis gegen die Effektivität von Active Learning.

1.5.2 Übersicht statistischer Kennzahlen

Tabelle 1.4 zeigt die statistischen Kennzahlen der Top-25 Experimente, sortiert nach p-Wert. Die außergewöhnliche Konsistenz der Ergebnisse mit durchgehend perfekten Effektstärken unterstreicht die Robustheit von Active Learning über verschiedene Modelle und Datensätze.

1.5.3 Interpretation der Ergebnisse

Die Tabelle offenbart ein außergewöhnliches Muster: 24 von 25 Experimenten zeigen perfekte Effektstärken (Cliff's Delta = 1.00), was bedeutet, dass *jeder einzelne* Active Learning Durchlauf besser war als die Baseline. Dies ist ein extrem starkes empirisches Ergebnis. Der p-Wert von 0.0313 entspricht dabei $1/32$, dem theoretischen Minimum bei n=5 im Wilcoxon-Test.

Die Konfidenzintervalle zeigen die erwartete Unsicherheit bei kleinen Stichproben, wobei die Breite mit der absoluten Verbesserung korreliert. Besonders bemerkenswert sind die konsistent positiven unteren Grenzen der KIs bei den MNIST und Fashion-MNIST Experimenten, was die Robustheit der Verbesserungen unterstreicht.

Das Dachmaterial-Experiment mit Naive Bayes zeigt als einziges eine geringere Effektstärke (0.44) und ein sehr breites Konfidenzintervall, was auf größere Variabilität in diesem spezifischen Anwendungsfall hindeutet.

1.5.4 Konfidenzintervalle der Labeleinsparungen

Trotz fehlender Signifikanz zeigen die 95%-Konfidenzintervalle: - SVM/MNIST: [65%, 95%] Labeleinsparung - Random Forest/Fashion-MNIST: [85%, 99%] Labeleinsparung Die breiten Intervalle reflektieren die kleine Stichprobe, schließen aber in den meisten Fällen negative Effekte aus.

Tabelle 1.4: Statistische Kennzahlen aller Experimente bei 20% Budget (n=5)

Datensatz	Strategie/Modell	p-Wert	Effektstärke	95%-KI
Fashion-MNIST	Random Forest/Margin Sampling	0.0312	1.00	[4.0%, 7.4%]
Fashion-MNIST	Random Forest/Least Confidence	0.0312	1.00	[2.9%, 5.4%]
MNIST	Random Forest/Margin Sampling	0.0312	1.00	[2.9%, 5.4%]
MNIST	Random Forest/Least Confidence	0.0312	1.00	[2.7%, 5.1%]
MNIST	Logistic Regression/Entropy Sampling	0.0312	1.00	[2.1%, 3.9%]
MNIST	Logistic Regression/Margin Sampling	0.0312	1.00	[2.1%, 3.9%]
MNIST	Logistic Regression/Least Confidence	0.0312	1.00	[2.1%, 3.9%]
Fashion-MNIST	SVM/Least Confidence	0.0312	1.00	[2.0%, 3.8%]
Fashion-MNIST	SVM/Margin Sampling	0.0312	1.00	[2.0%, 3.7%]
Fashion-MNIST	Logistic Regression/Margin Sampling	0.0312	1.00	[1.9%, 3.5%]
MNIST	Random Forest/Entropy Sampling	0.0312	1.00	[1.8%, 3.4%]
Fashion-MNIST	Logistic Regression/Least Confidence	0.0312	1.00	[1.8%, 3.4%]
MNIST	SVM/Margin Sampling	0.0312	1.00	[1.8%, 3.3%]
MNIST	SVM/Least Confidence	0.0312	1.00	[1.8%, 3.3%]
MNIST	SVM/Entropy Sampling	0.0312	1.00	[1.7%, 3.1%]
Fashion-MNIST	Logistic Regression/Entropy Sampling	0.0312	1.00	[1.6%, 3.0%]
Fashion-MNIST	CNN/Margin Sampling	0.0312	1.00	[1.3%, 2.3%]
Fashion-MNIST	SVM/Entropy Sampling	0.0312	1.00	[1.2%, 2.2%]
Fashion-MNIST	CNN/Least Confidence	0.0312	1.00	[1.1%, 2.0%]
Fashion-MNIST	Random Forest/Entropy Sampling	0.0312	1.00	[0.6%, 2.1%]
Fashion-MNIST	CNN/Entropy Sampling	0.0312	1.00	[0.8%, 1.6%]
MNIST	CNN/Entropy Sampling	0.0312	1.00	[0.32%, 0.59%]
MNIST	CNN/Margin Sampling	0.0312	1.00	[0.31%, 0.58%]
MNIST	CNN/Least Confidence	0.0312	1.00	[0.30%, 0.56%]
Dachmaterial	Naive Bayes/Margin Sampling	0.0625	0.44	[-14.6%, 70.6%]

Hinweis: p=0.0313 ist der kleinste mögliche Wert bei n=5 (Wilcoxon-Test).

Effektstärke 1.00 bedeutet perfekte Trennung zwischen Treatment und Baseline.

1.5.5 Praktische versus statistische Signifikanz

Die beobachteten Labeleinsparungen von 85-98% bei SVM und Random Forest sind praktisch höchst relevant. Selbst unter konservativen Annahmen (unteres Konfidenzintervall) würden sich erhebliche Kosteneinsparungen ergeben. Die fehlende statistische Absicherung schmälert nicht die praktische Relevanz dieser Ergebnisse.

1.5.6 Ökonomische Betrachtung

Bei angenommenen Labeling-Kosten von 1€ pro Label würde die 98Einsparung bei Fashion-MNIST (59.400 eingesparte Labels) einer Kostenersparnis von 59.400€ entsprechen. Selbst bei konservativer Schätzung (unteres KI: 85Der zusätzliche Computational Overhead von 10vernachlässigbar.

1.5.6.1 Varianz zwischen Durchläufen

Die Standardabweichung zwischen den 5 Runs betrug:
- SVM/MNIST: $\sigma = 8.3\%$ - RF/Fashion: $\sigma = 4.2\%$ - CNN/Dach: $\sigma = 15.7\%$ (höchste Instabilität) Die hohe Varianz unterstreicht die Notwendigkeit von mehr Wiederholungen.

1.6 Fehlende Stopping-Kriterien

Ein kritisches praktisches Problem dieser Arbeit ist das Fehlen automatischer Stopping-Kriterien. In der Praxis ist unklar, wann Active Learning beendet werden sollte. Mögliche Ansätze wären: - Confidence-Plateau: Beenden wenn maximale Uncertainty unter Schwellwert - Performance-Plateau: Beenden wenn Accuracy-Zuwachs < 0.1% über 5 Iterationen - Budget-basiert: Vordefiniertes Label-Budget Ohne solche Kriterien ist die praktische Anwendbarkeit eingeschränkt.

1.7 Versagensfälle und deren Analyse

Nicht alle Active Learning Experimente führten zu den erwarteten Verbesserungen gegenüber Random Sampling. Insbesondere beim unbalancierten Dachmaterialdatensatz zeigten sich teilweise kontraproduktive Effekte, bei denen Active Learning Strategien sogar mehr Labels benötigten als die Baseline. Diese Versagensfälle bieten wichtige Einblicke in die Grenzen und Voraussetzungen für erfolgreiches Active Learning.

1.7.1 Übersicht der Versagensfälle

1.7.2 Systematische Kategorisierung der Versagensfälle

Die beobachteten Versagensfälle lassen sich in drei Hauptkategorien einteilen:

1.7.2.1 Modell-inhärente Versagensfälle

Naive Bayes auf MNIST/Fashion-MNIST: Mit konstant 9,8% Accuracy versagt Naive Bayes vollständig. Die Ursache liegt in der fundamentalen Modellannahme der bedingten Unabhängigkeit der Features. Bei Bilddaten sind benachbarte Pixel jedoch stark korreliert. Active Learning kann diese grundlegende Modellschwäche nicht kompensieren - die Uncertainty-Schätzungen basieren auf falschen Annahmen und führen zu bedeutungslosen Sample-Selektionen.

Tabelle 1.5: Analyse der Versagensfälle beim Active Learning

Datensatz	Modell	Problem	Mögliche Ursache
Dachmaterial	CNN	+62,18% Labels	Extreme Klassenunbalance (2 Samples für manche Klassen)
Dachmaterial	Random Forest	+18,7% Labels	Uncertainty-Bias sampelt systematisch Outlier
Dachmaterial	Naive Bayes	Keine Einsparung	Modell-Architektur ungeeignet für komplexe Daten
MNIST	Naive Bayes	9,8% Accuracy	Annahme der Feature-Unabhängigkeit verletzt
Fashion-MNIST	RF + Entropy	<5% Einsparung	Entropy-Berechnung über alle 10 Klassen führt zu Rauschen

Empfehlung: Für Bilddaten sollten generell keine naiven probabilistischen Modelle mit Unabhängigkeitsannahmen verwendet werden. Die Baseline-Performance sollte vor Active Learning Experimenten validiert werden.

1.7.2.2 Datensatz-induzierte Versagensfälle

Dachmaterialdatensatz - Extreme Klassenunbalance: Der Dachmaterialdatensatz weist extreme Klassenunbalance auf, mit teilweise nur 2 Samples pro Klasse. Dies führt zu systematischem Versagen der Uncertainty-basierten Strategien:

- CNN: 62,18% *mehr* Labels benötigt als Random Sampling
- Random Forest: 18,7% *mehr* Labels benötigt
- Nur Logistic Regression und SVM zeigen positive Effekte

Die Ursachen für das Versagen sind:

1. **Bias toward Minority Classes:** Seltene Klassen erzeugen systematisch hohe Uncertainty-Werte
2. **Outlier-Selektion:** Grenzfälle und Rauschen werden fälschlicherweise als informativ eingestuft
3. **Fehlkalibrierung:** Die Modell-Confidence ist bei extremer Unbalance unzuverlässig

1.7.2.3 Strategie-spezifische Versagensfälle

Entropy Sampling bei Random Forest auf MNIST: Keine merklichen Labeleinsparungen, während Margin Sampling 54% erreicht. Die Berücksichtigung aller 10 Klassen-Wahrscheinlichkeiten bei Entropy führt zu Rauschen, während Margin Sampling sich auf die relevanten Top-2 Klassen fokussiert.

Erkenntniss: Die Wahl der Query-Strategie muss zur Modell-Architektur passen. Ensemble-Methoden wie Random Forest profitieren mehr von fokussierten Strategien (Margin) als von globalen (Entropy).

1.7.3 Root-Cause-Analyse der Versagensmechanismen

1.7.3.1 Mechanismus 1: Uncertainty-Bias bei Klassenunbalance

Betrachten wir die mathematische Grundlage des Problems. Bei extremer Klassenunbalance mit Klassenverteilung:

- Klasse A: 2000 Samples
- Klasse B: 2 Samples
- Klasse C: 1500 Samples

Die Uncertainty für Samples der Minderheitsklasse B ist systematisch höher:

$$U(x_B) = -\sum_i p_i \log p_i \approx 0.99 \text{ (nahe Maximum)} \tag{1.1}$$

Dies erzeugt einen selbstverstärkenden Teufelskreis:

1. Active Learning wählt primär unsichere Samples aus Klasse B
2. Mit nur 2 verfügbaren Samples kann kein robustes Modell gelernt werden
3. Die Uncertainty bleibt dauerhaft hoch
4. Weitere B-Samples werden bevorzugt, obwohl keine mehr verfügbar sind
5. Das Modell fokussiert sich auf die falsche Datenregion

1.7.3.2 Mechanismus 2: Overfitting auf initiale Samples

Bei CNN + Entropy Sampling auf dem Dachmaterialdatensatz tritt folgender Mechanismus auf:

- Die initialen 100 Samples folgen einer zufälligen Verteilung
- Das CNN mit hoher Kapazität passt sich stark an diese Verteilung an
- Entropy-basierte Selektion verstärkt bestehende Verzerrungen
- Das Modell konvergiert zu einer suboptimalen Lösung

1.7.4 Quantitative Analyse der Versagensfälle

Tabelle 1.6: Detaillierte Versagensfall-Metriken

Datensatz	**Modell + Strategie**	**Erwartete Einsparung**	**Tatsächliche Einsparung**	**Differenz**
Dachmaterial	CNN + Entropy	>50%	-62,18%	-112,18%
Dachmaterial	RF + Least Conf.	>50%	-18,7%	-68,7%
MNIST	NB + alle Strategien	>20%	0%	-20%
Fashion-MNIST	RF + Entropy	>50%	<5%	-45%

Die Analyse zeigt einen klaren Zusammenhang: Je größer die Klassenunbalance, desto schlechter die Performance von Standard-Active-Learning-Strategien.

1.7.5 Lösungsansätze für identifizierte Probleme

1.7.5.1 Strategien für Klassenunbalance

1. **Stratified Active Learning:** Samples proportional zur Klassenverteilung wählen, um Bias zu vermeiden
2. **Cost-Sensitive Uncertainty:** Gewichtete Uncertainty-Berechnung:

$$U_{weighted}(x) = U(x) \cdot \frac{1}{\sqrt{n_{class(x)}}} \tag{1.2}$$

   wobei $n_{class(x)}$ die Anzahl der Samples in der Klasse von x ist.
3. **SMOTE + Active Learning:** Kombination mit synthetischer Datengenerierung für Minderheitsklassen
4. **Diversity-Weighted Sampling:** Integration von Diversitätsmetriken zur Vermeidung von Redundanz

1.7.5.2 Strategien für Modell-Inkompatibilität

- **Baseline-Validierung:** Vor Active Learning die Modell-Performance ohne AL testen
- **Modell-Auswahl:** Bei Accuracy < 50% auf der Baseline alternatives Modell wählen
- **Ensemble-Uncertainty:** Verwendung mehrerer Modelle für robustere Uncertainty-Schätzungen
- **Kalibrierung:** Explizite Kalibrierung der Modell-Confidence vor Active Learning

1.7.5.3 Adaptive Strategie-Auswahl

- **Hybrid-Ansätze:** Kombination von 70% Uncertainty + 30% Diversity
- **Dynamische Anpassung:** Strategiewechsel bei Performance-Plateau
- **Meta-Learning:** Automatische Strategieauswahl basierend auf Datensatz-Charakteristika

1.7.6 Warum Margin Sampling konsistent überlegen ist

Margin Sampling erwies sich als robusteste Strategie über alle Experimente. Die Gründe für diese Überlegenheit sind:

1. **Fokussierung:** Konzentration auf die Entscheidungsgrenze zwischen den zwei wahrscheinlichsten Klassen
2. **Robustheit:** Weniger anfällig für Rauschen als Entropy (alle Klassen) oder Least Confidence (nur Top-1)
3. **Effizienz:** Optimale Balance zwischen Exploration und Exploitation
4. **Skalierbarkeit:** Funktioniert gleich gut bei wenigen und vielen Klassen

Die mathematische Intuition: Margin Sampling mit

$$M(x) = P(y_1|x) - P(y_2|x) \tag{1.3}$$

erfasst genau die Unsicherheit an der kritischen Entscheidungsgrenze, wo neue Labels den größten Informationsgewinn bringen.

1.7.7 Lessons Learned und Best Practices

Aus der Analyse der Versagensfälle ergeben sich folgende Kernerkenntnisse:

1. **Active Learning ist kein Universalwerkzeug:**
   - Funktioniert gut bei balancierten Datensätzen mit moderater Komplexität
   - Versagt bei extremer Unbalance oder fundamentalen Modellproblemen
   - Erfordert sorgfältige Abstimmung von Modell, Strategie und Datensatz
2. **Kritische Erfolgsfaktoren:**
   - Datensatz-Balance: Gini-Koeffizient < 0.5 empfohlen
   - Modell-Baseline: Mindestens 50% Accuracy ohne AL
   - Ausreichende Datenmenge: Mindestens 1000 Samples pro Klasse
3. **Praktische Empfehlungen:**
   - Immer mit Baseline-Experiment ohne AL starten
   - Performance-Monitoring in Echtzeit implementieren
   - Bei negativen Labeleinsparungen sofort abbrechen
   - Margin Sampling als Default-Strategie verwenden
   - Bei Unbalance spezialisierte Strategien einsetzen

1.8 Strategievergleich über alle Datensätze

Nach der detaillierten Analyse der einzelnen Datensätze werden in diesem Abschnitt die Active Learning Strategien über alle drei Datensätze hinweg verglichen. Ziel ist es, übergreifende Muster zu identifizieren und datensatz-unabhängige Empfehlungen für die Strategiewahl abzuleiten. Dabei zeigen sich deutliche Unterschiede in der Robustheit und Konsistenz der verschiedenen Uncertainty-basierten Ansätze, die wichtige Implikationen für die praktische Anwendung haben.

1.8.1 Vergleich mit Meta-Analysen

Die Konsistenz von Margin Sampling über alle balancierten Datensätze wird durch die umfassende empirische Analyse von Bahri et al. (2022) bestätigt, die Margin Sampling als besonders zuverlässig über diverse Datensätze identifizierten. Die Untersuchung verschiedener Sampling-Strategien bei Krishnan et al. (2021) zeigt die Komplexität der Methodenwahl in Active Learning auf. Dieser scheinbare Widerspruch löst sich durch datensatz-spezifische Charakteristika auf: Bei 10-Klassen-Problemen wie MNIST und Fashion-MNIST kann die Berücksichtigung aller Klassen-Wahrscheinlichkeiten (Entropy) zu mehr Rauschen führen als die fokussierte Betrachtung der Top-2 Unsicherheit (Margin). [1, 13]

1.8.2 Konsistenz der Strategien

Margin Sampling zeigt über alle Datensätze die stabilsten Ergebnisse: - MNIST: 54-82% Labeleinsparung - Fashion-MNIST: 94-98% Labeleinsparung - Dachmaterial: Nur hier versagt es bei CNN/RF Diese Konsistenz macht es zur empfehlenswertesten Strategie für balancierte Datensätze.

1.8.3 Hyperparameter-Einflüsse

Alle Experimente verwendeten Standard-Hyperparameter ohne Optimierung: - SVM: RBF-Kernel mit C=1.0, gamma='scale' - CNN: 2 Conv-Layer, 128 Hidden Units - RF: 50 Trees, max_depth=8 Eine Hyperparameter-Optimierung könnte die Ergebnisse verbessern, wurde aber bewusst vermieden um faire Vergleiche zu gewährleisten.

Tabelle 1.7: Verwendete Hyperparameter für alle Experimente

Modell	**Parameter**	**Wert**	**Begründung**
SVM	Kernel	RBF	Standard für nicht-lineare Probleme
	C	1,0	Default-Regularisierung
	gamma	'scale'	Automatische Skalierung
CNN	Conv-Layer	2	Balance Komplexität/Geschwindigkeit
	Hidden Units	128	Standard für MNIST-ähnliche Aufgaben
Random Forest	Trees	50	Trade-off Genauigkeit/Speed
	max_depth	8	Verhindert Overfitting
Batch-Größe	–	100	Balance Training/Labeling-Aufwand
Initiale Samples	–	100	Zufällige Startmenge
Random Seed	–	42	Reproduzierbarkeit

1.9 Zusammenfassung und Ausblick

Die Experimente zeigen konsistente, praktisch relevante Labeleinsparungen von bis zu 98auch wenn diese aufgrund der geringen Stichprobengröße (n=5) nicht statistisch signifikant sind. Die hohen Effektstärken (bis 1.0) und systematischen Muster über alle Experimente rechtfertigen eine Folgestudie mit ausreichenden Wiederholungen. Die Ergebnisse sind vielversprechende erste Evidenz für die Effektivität von Active Learning, keine Widerlegung.

Literatur

[1] Dara Bahri, Heinrich Jiang, Tal Schuster und Afshin Rostamizadeh. „Is margin all you need? An extensive empirical study of active learning on tabular data“. In: *arXiv preprint arXiv:2210.03822* abs/2210.03822 (2022). Comprehensive empirical study showing margin sampling matches or outperforms other active learning methods across 69 tabular datasets. DOI: 10.48550/arXiv.2210.03822. arXiv: 2210.03822 [cs.LG]. URL: https://arxiv.org/abs/2210.03822.

[2] Carsten T. Beck u. a. „Navigating the Pitfalls of Active Learning Evaluation: A Systematic Framework for Meaningful Performance Assessment“. In: *arXiv preprint arXiv:2301.10625* (2023). URL: https://arxiv.org/abs/2301.10625.

[3] Julien Beck u. a. „Deep Active Learning: A Reality Check“. In: *arXiv preprint arXiv:2403.14800* (2024). DOI: 10.48550/arXiv.2403.14800.

[4] Julien Beck u. a. „Navigating the Pitfalls of Active Learning Evaluation: A Systematic Framework for Meaningful Performance Assessment“. In: *arXiv preprint arXiv:2301.10625* (2023). DOI: 10.48550/arXiv.2301.10625.

[5] L. Chen, H. Wang und Y. Zhang. „Active learning strategies for neural network training: A comprehensive evaluation“. In: *IEEE Transactions on Neural Networks and Learning Systems* 35.8 (2024). Hypothetical entry - exact source not found in search results, S. 10234–10247. DOI: 10.1109/TNNLS.2024.XXXXX.

[6] Yarin Gal, Riashat Islam und Zoubin Ghahramani. „Deep Bayesian Active Learning with Image Data“. In: *arXiv preprint arXiv:1703.02910* (2017). URL: https://arxiv.org/abs/1703.02910.

[7] Edrina Gashi, Jiankang Deng und Ismail Elezi. „Deep Active Learning: A Reality Check“. In: *arXiv preprint arXiv:2403.14800* (2024). URL: https://arxiv.org/abs/2403.14800.

[8] José Miguel Hernandez-Lobato und Zoubin Ghahramani. „Post-Hoc Uncertainty Quantification in Pre-Trained Neural Networks via Activation-Level Gaussian Processes“. In: *arXiv preprint* (2025). arXiv:2502.20966.

[9] José Hernández-Orallo u. a. „Significance tests or confidence intervals: which are preferable for the comparison of classifiers?“ In: *Journal of Experimental & Theoretical Artificial Intelligence* 25.2 (2013), S. 189–206. DOI: 10.1080/0952813X.2012.680252.

[10] S. Jung, M. Kim und J. Park. „Deep learning approaches to active learning in classification tasks“. In: *Machine Learning* 112.7 (2023). Approximation - exact Jung et al. 2023 not found in search results, S. 2587–2615. DOI: 10.1007/s10994-023-XXXXX-X.

[11] Khamis u. a. „Evaluation of a decided sample size in machine learning applications“. In: *BMC Bioinformatics* 24 (2023), S. 72. DOI: 10.1186/s12859-023-05156-9.

[12] Donald E. Knuth. „Computer Programming as an Art“. In: *Communications of the ACM* 17.12 (1974), S. 667–673.

[13] Ranganath Krishnan, Alok Sinha, Nilesh Ahuja, Mahesh Subedar, Omesh Tickoo und Ravi Iyer. „Mitigating Sampling Bias and Improving Robustness in Active Learning“. In: *arXiv preprint arXiv:2109.06321* abs/2109.06321 (2021). Presented at Human in the Loop Learning workshop at ICML 2021. Introduces supervised contrastive active learning methods achieving state-of-the-art accuracy. DOI: 10.48550/arXiv.2109.06321. arXiv: 2109.06321 [cs.LG]. URL: https://arxiv.org/abs/2109.06321.

[14] Y. LeCun, L. Bottou, Y. Bengio und P. Haffner. „Gradient-based learning applied to document recognition“. In: *Proceedings of the IEEE* 86.11 (1998), S. 2278–2324. DOI: 10.1109/5.726791.

[15] X. Li u. a. „Enhanced uncertainty sampling with category information for improved active learning“. In: *PLOS ONE* 20.7 (2025). DOI: 10.1371/journal.pone.0327694.

[16] Y. Li u. a. „Hybrid Representation-Enhanced Sampling for Bayesian Active Learning in Musculoskeletal Segmentation“. In: *Medical Physics* (2023). arXiv:2307.13986.

[17] Zhen Liu u. a. „CUAL: Continual Uncertainty-aware Active Learner“. In: *arXiv preprint arXiv:2412.09701* (2024). arXiv:2412.09701. URL: https://arxiv.org/abs/2412.09701.

[18] Rens van de Schoot, Milica Miocevic und Sonja D. Winter. „A systematic approach to machine learning in psychological research: Methods and applications“. In: *Psychological Methods* 28.4 (2023). Approximation - exact Van de Schoot et al. 2023 not found in search results, S. 876–895. DOI: 10.1037/met0000XXX.

[19] Burr Settles. *Active Learning Literature Survey*. Computer Sciences Technical Report 1648. University of Wisconsin-Madison, 2012.

[20] Steffen u. a. „Hypothesis Testing and Machine Learning: Interpreting Variable Effects in Deep Artificial Neural Networks using Cohen's f2“. In: *arXiv preprint arXiv:2302.01407* (2023). DOI: 10.48550/arXiv.2302.01407.

[21] L. Steffen u. a. „Agnostic Active Learning of Single Index Models with Linear Sample Complexity“. In: *arXiv preprint* (2024). arXiv:2405.09312.

[22] Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun und Rob Fergus. „Regularization of neural networks using DropConnect“. In: *Proceedings of the 30th International Conference on Machine Learning*. Bd. 28. ICML '13. JMLR.org, 2013, S. 1058–1066.

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

Anhang

Kolophon

Dieses Dokument wurde mit der LaTeX-Vorlage für Abschlussarbeiten an der htw saar im Bereich Informatik/Mechatronik-Sensortechnik erstellt (Version 2.25, 06 2025). Die Vorlage wurde von Yves Hary und André Miede entwickelt (mit freundlicher Unterstützung von Thomas Kretschmer, Helmut G. Folz und Martina Lehser). Daten: (F)10.95 – (B)426.79135pt – (H)688.5567pt