

Konsortialschlussbericht 2025

Verbundvorhaben



AutoQML – Developer-Suite für automatisiertes maschinelles Lernen mit Quantencomputern

Gefördert durch das Bundesministerium für Wirtschaft und Klimaschutz (BMWK)

Konsortialführer:		Förderkennzeichen:
	Fraunhofer IAO	01MQ22002B
Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO		
Projektleiter:		Tel.:
Dr. Christian Tutschku		+49 176 82202000
Laufzeit des Vorhabens:		Fax:
von: 01.01.2022 bis: 31.12.2024		
Berichtszeitraum des Vorhabens:		E-mail:
von: 01.01.2022 bis: 31.12.2024		christian.tutschku@iao.fraunhofer.de

Projektpartner:



**Fraunhofer-Institut für Produktionstechnik und Automatisierung
IPA**

Prof. Dr.-Ing. Marco Huber, Dr. Marco Roth
Dr. David Kreplin



IAV GmbH Ingenieurgesellschaft Auto und Verkehr
Dr. Peter Schichtel



USU Software AG
Dr. Andreas Röpnack



KEB Automation KG
Khaled Al-Gumaei



TRUMPF SE + Co. KG (Holding)
Dr. Frederick Struckmeier



Zeppelin GmbH
Horst Stühler



GFT Integrated Systems GmbH
Henning Krause

Inhaltsverzeichnis

I. Kurze Darstellung zu	4
Aufgabenstellung	4
Voraussetzungen, unter denen das Vorhaben durchgeführt wurde,	4
Planung und Ablauf des Vorhabens	5
Wissenschaftlichem und technischem Stand, an den angeknüpft wurde	6
Zusammenarbeit mit anderen Stellen	7
II. Eingehende Darstellung	7
Verwendung der Zuwendung und des erzielten Ergebnisses im Einzelnen, mit Gegenüberstellung der vorgegebenen Ziele, wichtigsten Positionen des zahlenmäßigen Nachweises und Notwendigkeit und Angemessenheit der geleisteten Arbeit.	7
AP 1: Anwendungsfelder und Anforderungen	7
AP 2: Methodik für automatisiertes Machine Learning mit Quantencomputern	8
AP 3: Algorithmen Quantum Machine Learning	10
AP 4: AutoQML Developer-Suite: Werkzeuge und Komponenten	11
AP 5: Applikationen: Produktion	13
AP 6: Applikationen: Automotive.....	16
Benchmarking AP5 & AP6.....	18
Fortschreibung des Verwertungsplans.	19
Während der Durchführung des Vorhabens dem ZE bekannt gewordener Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen,.....	21
Erfolgte oder geplante Veröffentlichungen des Ergebnisses nach Nr.11.	21

I. Kurze Darstellung zu

Aufgabenstellung

Im Allgemeinen verspricht QC (Quantencomputing) bestimmte Probleme um Größenordnungen (zumeist polynomisch bis hin zu exponentiell) schneller zu lösen als es mittels konventioneller Algorithmen der Fall ist. Das gilt im Speziellen auch für den Unterbereich QML (Quantenmaschinelles Lernen) der sich damit befasst, wie dieser Quantenvorteil für derzeitige ML-Anwendungen nutzbar gemacht werden kann. Allerdings ist in diesem Bereich eine ähnliche Entwicklung wie jüngst im Feld ML (maschinelles Lernen) zu erwarten. Durch die Neuartigkeit der Technologie und das kaum verfügbare interdisziplinäre Fachwissen ist ein gravierender Fachkräftemangel absehbar. Inspiriert von den Automatisierungsbestrebungen im Feld ML, ist es Ziel dieses Projekts eine ähnliche Automatisierung für QML zu entwickeln.

Ein wichtiger Baustein für die Entwicklung von Lösungen für Quantencomputer sind also Methoden und Werkzeuge, die die Komplexität der zugrundeliegenden quantenmechanischen Prozesse abstrahieren und aggregieren. Ein naheliegender Ansatz ist es, die Entwicklung im Bereich ML zum Vorbild zu nehmen und auf QC zu übertragen. Dort wird versucht, mittels Automated Machine Learning (AutoML), diesen Flaschenhals durch Automatisierung von ML-Prozessen zu überwinden. Insbesondere wird dabei die Wahl der konkreten ML-Algorithmen, sowie die zugehörige Hyperparameteroptimierung automatisiert. Durch diese Vereinfachung ist es auch konventionellen Entwicklern ohne tiefgehende ML-Expertise möglich, geeignete Probleme mit ML-Methoden zu lösen und so den Mehrwert dieser zukunftsweisenden Technologie zu nutzen. Ziel des Projektes AutoQML ist es, diese Methodik auf Quantum Machine Learning (QML) zu übertragen.

Die beiden Fraunhofer Institute zeichnen hierbei für den wissenschaftlichen Aspekt der verwendeten QML-Methoden verantwortlich.

Um dies möglichst anwendungsnah durchzuführen, erfolgt die Entwicklung geleitet durch vier Anwendungsfälle aus den Bereichen Automotive und Produktion, die von den jeweiligen Industriepartnern eingebracht werden. Diese werden in Zusammenarbeit mit den Fraunhofer-Instituten mittels ML und QML gelöst und die dabei entstandenen Methoden werden dann im zu konstruierenden AutoQML Framework vereint. Die Architektur und Implementierung des Frameworks erfolgen in Zusammenarbeit der beiden Softwarehäuser und der beiden Fraunhofer-Institute.

Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Sowohl das Fraunhofer IAO als auch das Fraunhofer IPA haben sich in der Vergangenheit herausragende Expertise und Erfahrung in der Entwicklung hybrider Quantensoftware erarbeitet. Beispielsweise spielt das Fraunhofer IAO eine führende Rolle im Kompetenzzentrum Quantencomputing Baden-Württemberg und erarbeitete in einem der beiden größten Teilprojekte des Kompetenzzentrums, dem Projekt SEQUOIA, gemeinsam mit dem Fraunhofer IPA Methoden und Werkzeuge zum hybriden Quanten-Software-Engineering. Außerdem leiten die beiden Fraunhofer-Institute IAO und IPA zusammen seit Jahren das KI-Fortschrittszentrum »Lernende Systeme und Kognitive Robotik«¹, welches Partner im Cyber-Valley² ist, der größten Forschungs Kooperation im Bereich KI. Ergebnisse und Knowhow dieser beiden Projekte wurden in das Projekt AutoQML zielführend eingebracht und anwenderorientiert im Rahmen des Projektvorhabens zur Marktreife weiterentwickelt. Obgleich im Verbundforschungsprojekt SEQUOIA keine QML Methoden schwerpunktmäßig untersucht worden sind, wurden hier fundierte Kenntnisse im hybriden Quantencomputing erarbeitet, die zielführend in das AutoQML Projekt eingebracht wurden. Beispiele dafür sind: Simulation von Quantenalgorithmen auf (hauseigenen) GPU-Servern, Testen Von Fehlermodellen, Erfahrung mit realen NISQ Backends (IBM Q System One Ehningen, etc.), Erfahrung mit variationellen Algorithmen (QAOA, VQE, etc.), Erfahrung mit kombinatorischen Optimierungsproblemen (Grover, etc.), Erfahrung mit Daten En- und Decoding (Messprozesse), Erfahrung mit Fehlermitigation, Erfahrung mit hybridem Quantensoftware Engineering.

¹ <https://www.ki-fortschrittszentrum.de/>

² <https://cyber-valley.de/>

Im Bereich AutoML verfügt das Fraunhofer IPA über umfassende Kenntnisse, die in einer aktuellen Übersichtsarbeit zum Thema zusammengefasst sind.³ Weiterhin hat das IPA bereits tiefergehende Arbeiten zur Lösung der Pipelinesuche durchgeführt.⁴ Das Projekt wird am IPA durch das Zentrum für Cyber Cognitive Intelligence (CCI)⁵ durchgeführt, das vom Land Baden-Württemberg gefördert wird. Das CCI ist ein industrienaher Forschungs- und Entwicklungspartner für die Umsetzung von Applikationen im Bereich Künstliche Intelligenz (KI) und insbesondere ML in der produzierenden Industrie. Ziel des Zentrums ist es, sowohl die KI-Forschung als auch den Technologietransfer von KI und ML in die Anwendung voranzutreiben. Das IPA ist ferner an dem vom Bundesministerium für Bildung und Forschung geförderten Wasserstoffleitprojekt H2GIGA⁶ beteiligt. Im zugehörigen Teilprojekt Degrad-QML-AEL werden Degradationsmechanismen unterschiedlicher Brennstoffzellen untersucht. Für die Degradationsanalyse werden Quantenalgorithmen eingesetzt. Der Fokus liegt hier sowohl auf der Simulation von quantenchemischen Reaktionen als auch auf der Optimierung von Elektrolyseprozessen und der Optimierung der Lebensdauer der Elektrolyseure.

Planung und Ablauf des Vorhabens

Im Folgenden ermöglicht die tabellarische Übersicht über die Arbeitspakete des Gesamtvorhabens in Tabelle 1 eine erste Orientierung. In der eingehenden Darstellung der Ergebnisse weiter unten wird der Ablauf jedes Arbeitspakets (AP) im Detail beschrieben. Daher soll hier nur ein Kurzüberblick gegeben werden. Es kann aber schon festgehalten werden, dass der tatsächliche Ablauf in allen wesentlichen Punkten der Planung entsprochen hat.

Das zentrale AP 4 „AutoQML Developer Suite“ dreht sich um die Schaffung des Endprodukts, des AutoQML Frameworks. In den Applikations-Arbeitspaketen AP 5 „Produktion“ und AP 6 „Automotive“ werden die Anwendungsfälle gelöst, dabei gewonnene Erkenntnisse und Methoden fließen in AP 4 ein. AP 1 „Anwendungsfelder und Anforderungen“ schafft mit der Anforderungsanalyse der Anwendungsfälle die Basis für AP 5 und AP 6. AP 2 „Methodik für automatisiertes ML mit QC“ schafft die theoretische Grundlage für den Frameworkbau in AP 4, unter anderem wurde dort das unterliegende AutoML Framework Ray ausgewählt. AP 3 „Algorithmen Quantum Machine Learning“ schafft die theoretische Grundlage zur Lösung der Anwendungsfälle mit QML-Methoden in AP 5 und AP 6. In Tabelle 1 kann man anhand des Gantt-Diagramms die oben geschilderten Abhängigkeiten nachvollziehen.

Arbeitspaket (AP)		1. Jahr												2. Jahr												3. Jahr												PM	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	ges.	
1	Anwendungsfelder und Anforderungen (GFT IN)																																						9
1.1	Charakterisierung, Klassifikation und Bewertung der Anwendungsfälle	x	x	x																																		2	
1.2	Anforderungen und Entwurf der Applikationen													x	x	x											x	x	x									2	
1.3	Datenverständnis und -vorbereitung													x	x	x											x	x	x									5	
2	Methodik für automatisiertes ML mit QC (IPA)																																					25	
2.1	Erweiterung des CRISP-DM Prozesses mit Quantencomputing													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	6		
2.2	Quantum Data Science													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	10		
2.3	Entscheidungs- und Migrationsprozesse konventionell/hybrid/QC													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	9		
3	Algorithmen Quantum Machine Learning (IAO)																																					43	
3.1	Untersuchung und Prototyping bestehende QML-Algorithmen													x	x	x	x	x	x	x	x	x	x														16		
3.2	Hybride Machine-Learning Algorithmen													x	x	x	x	x	x	x	x	x	x														18		
3.3	Vergleich und Auswahl													x	x	x	x	x																			9		
4	AutoQML Developer-Suite: Werkzeuge und Komponenten (USU)																																					34	
4.1	Schnittstellen und Einordnung in das AutoML-Ökosystem													x	x	x	x	x	x	x	x	x	x														5		
4.2	Werkzeuge und Komponenten													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	4		
4.3	QC-Hyperparameteroptimierung													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	20		
4.4	Durchgängige und dynamische Werkzeugketten													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	5		
5	Applikationen: Produktion (Trumpf)																																					21	
5.1	Umsetzung Klassisches Machine Learning													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	3		
5.2	Anwendung von QML auf Use-Cases													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	11		
5.3	End-to-end Umsetzung der Use-Cases mit QML													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	4		
5.4	Vergleich, Benchmarking und Evaluierung																																					3	
6	Applikationen: Automotive (Zeppelin)																																					21	
6.1	Umsetzung Klassisches Machine Learning													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	3		
6.2	Anwendung von QML auf Use-Cases													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	11		
6.3	End-to-end Umsetzung der Use-Cases mit QML													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	4		
6.4	Vergleich, Benchmarking und Evaluierung																																					3	
7	Transfer und Partnerkooperation (IAO)																																					13	
7.1	Öffentlichkeitsarbeit													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	5		
7.2	Partnerkooperation und Interessenten													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	5		
7.3	Integration mit PlanQK													x	x	x											x	x	x	x	x	x	x	x	x	x	x	3	
8	Projektmanagement (IAO)													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	14		
Summe														M1				M2								M3								M4					180

Tabelle 1: Detaillierter Zeitplan und konkrete Meilensteinplanung des Gesamtvorhabens

³ Zöller, Marc-André, and Marco F. Huber. "Benchmark and survey of automated machine learning frameworks." Journal of Artificial Intelligence Research 70 (2021): 409-472.

⁴ Marc-André Zöller, Tien-Dung Nguyen und Marco F. Huber, „Incremental Search Space Construction for Machine Learning Pipeline Synthesis“. In: Symposium on Intelligent Data Analysis (IDA), April 2021

⁵ https://www.ipa.fraunhofer.de/de/ueber_uns/Leitthemen/ki.html

⁶ <https://www.wasserstoff-leitprojekte.de/leitprojekte/h2giga>

Wissenschaftlichem und technischem Stand, an den angeknüpft wurde

Im Bereich des AutoML existieren unterschiedliche Ansätze, um die Optimierung von Hyperparametern zu automatisieren. Zöller und Huber (2021) bieten hierzu einen Überblick⁷. Der naive Ansatz der Rastersuche skaliert schlecht mit der Größe des Parameterraumes und eventuelle hierarchische Hyperparameterstrukturen werden nicht berücksichtigt. Der Random Search Ansatz skaliert besser, nutzt aber ebenfalls das Wissen über erfolgsversprechende

Regionen des Hyperparameterraumes nicht aus. Die Optimierung der Hyperparameter kann via Bayesscher Optimierung erfolgen. Hier wird eine Wahrscheinlichkeitsverteilung über den Raum der Hyperparameter geschätzt, um die optimalen Parameter wählen zu können. Insgesamt ist die Hyperparameteroptimierung mit den genannten Methoden sehr rechen- und hardwareintensiv. Teilziel des Vorhabens ist es, die Suche der Hyperparameter mittels Optimierung auf dem Quantencomputer zu beschleunigen.

Im Allgemeinen bieten QML-Methoden perspektivisch Vorteile gegenüber herkömmlichen ML Methoden. So sind insbesondere langfristig über die Integration geeigneter Quantensubroutinen Geschwindigkeitsvorteile zu erwarten.⁸ Weiterhin wurde gezeigt, dass Quantencomputer bestimmte Probleme klassifizieren können, die für klassische Computer nicht vom Zufall zu unterscheiden sind.⁹ Die dort verwendete Methode ist Teil einer allgemeineren Klasse von Algorithmen, die einen soliden Anhaltspunkt für die Generierung weiterer Vorteile bietet.¹⁰

Derzeitige Hardwarerealisierungen von Quantencomputern unterliegen technischen Einschränkungen. Zu Beginn des Projekts hatten aktuelle Quantencomputer nur ca. 50-100 Qubits mit einer deutlich beschränkten Kohärenzzeit. Diese Quantencomputer werden als NISQ¹¹ Computer bezeichnet. Langfristig werden viele der entwickelten Quantenalgorithmen nur über fehlerkorrigierte Hardware ausführbar sein. Um auch kurz- und mittelfristig bereits Quantencomputer nutzbar zu machen, stehen derzeit variationelle Algorithmen im Fokus. Hierbei handelt es sich um hybride Methoden, bei denen parametrisierte Quantenschaltkreise über eine klassische Optimierungsschleife auf die jeweilige Anwendung zugeschnitten werden. Im Bereich QML wurden hier schon vielversprechende Ansätze verfolgt und in Experimenten demonstriert.^{12,13}

Für die Lösung kombinatorischer Optimierungsprobleme wurden bereits einige variationelle Quantenalgorithmen entwickelt. Von besonderer Bedeutung ist hier der "Quantum Approximate Optimization Algorithm" (QAOA).¹⁴ Dieser wurde bereits mehrfach experimentell umgesetzt¹⁵ und bietet, mathematische Leistungsgarantien. Es wurde weiterhin gezeigt, dass QAOA-Ansätze im Allgemeinen nicht klassisch simulierbar sind¹⁶, was sie zu vielversprechenden Kandidaten für Quantenvorteile macht. Weiterhin wurden Varianten des Groveralgorithmus entwickelt, die einen polynomischen Geschwindigkeitsvorteil gegenüber klassischen Lösungen bieten und somit perspektivisch vielversprechend für die Lösung von kombinatorischen Optimierungsproblemen geeignet sind.¹⁷

⁷ Zöller, Marc-André, and Marco F. Huber. "Benchmark and survey of automated machine learning frameworks." *Journal of Artificial Intelligence Research* 70 (2021): 409-472.

⁸ Gilyén, András; Su, Yuan; Low, Guang Hao; Wiebe, Nathan (2019): Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics, S. 193–204. DOI: 10.1145/3313276.3316366.

⁹ Liu, Yunchao; Arunachalam, Srinivasan; Temme, Kristan (2021): A rigorous and robust quantum speed-up in supervised machine learning. In: *Nat. Phys.* DOI: 10.1038/s41567-021-01287-z.

¹⁰ Glick, Jennifer R.; Gujarati, Tanvi P.; Corcoles, Antonio D.; Kim, Youngseok; Kandala, Abhinav; Gambetta, Jay M.; Temme, Kristan (2021): Covariant quantum kernels for data with group structure. Online verfügbar unter <http://arxiv.org/pdf/2105.03406v1>

¹¹ Preskill, John (2018): Quantum Computing in the NISQ era and beyond. In: *Quantum* 2, S. 79. DOI: 10.22331/q-2018-08-06-79.

¹² Cerezo, M.; Arrasmith, Andrew; Babbush, Ryan; Benjamin, Simon C.; Endo, Suguru; Fujii, Keisuke et al. (2021): Variational quantum algorithms. In: *Nat Rev Phys.* DOI: 10.1038/s42254-021-00348-9.

¹³ Bharti, Kishor; Cervera-Lierta, Alba; Kyaw, Thi Ha; Haug, Tobias; Alperin-Lea, Sumner; Anand, Abhinav et al. (2021): Noisy intermediate-scale quantum (NISQ) algorithms. Online verfügbar unter <http://arxiv.org/pdf/2101.08448v1>

¹⁴ Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. arXiv:1411.4028 (2014).

¹⁵ z.B. Harrigan, Matthew P.; Sung, Kevin J.; Neeley, Matthew; Satzinger, Kevin J.; Arute, Frank; Arya, Kunal et al. (2021): Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. In: *Nat. Phys.* 17 (3), S. 332–336. DOI: 10.1038/s41567-020-01105-y.

¹⁶ Farhi, E. & Harrow, A. W. Quantum supremacy through the quantum approximate optimization algorithm. arXiv preprint arXiv:1602.07674 (2016).

¹⁷ Gilliam, Austin; Woerner, Stefan; Goniçulea, Constantin (2021): Grover Adaptive Search for Constrained Polynomial Binary Optimization. In: *Quantum* 5, S. 428. DOI: 10.22331/q-2021-04-08-428.

Zusammenarbeit mit anderen Stellen

Unterauftrag HQS

Im Jahr 2023 entwickelte die HQS Quantum Simulation GmbH eine automatisierte Auswahl der IBM Quantenbackends für QML-Algorithmen. Die automatisierte Auswahl nutzt die Mapomatic-Bibliothek und das HQAA von HQS und bietet zwei Modi: Qualität und Geschwindigkeit. Der Qualität-Modus optimiert Fehlerquoten und Leistung, während der Geschwindigkeitsmodus die schnellste Ausführung priorisiert. Beide Modi bewerten die erwartete Fehlerquote. Zusätzlich wurden parallelisierte Qiskit Sampler und Estimatoren entwickelt, um Schaltkreise effizienter auf größerer Quantenhardware auszuführen. Beide Funktionen wurden in die sQUlearn-Bibliothek integriert und sind darüber in das AutoQML Framework eingebunden.

II. Eingehende Darstellung

Verwendung der Zuwendung und des erzielten Ergebnisses im Einzelnen, mit Gegenüberstellung der vorgegebenen Ziele, wichtigsten Positionen des zahlenmäßigen Nachweises und Notwendigkeit und Angemessenheit der geleisteten Arbeit.

Wesentliches technisches Ergebnis des Projekts ist das hier geschaffene Software-Framework „AutoQML“. In seine Entwicklung sind wissenschaftliche Erfahrungen aus allen vier behandelten Anwendungsfällen eingeflossen. Dabei wurden Erkenntnisse über den Nutzen von quantenmaschinellern Lernen in den Bereichen Klassifikation, Regression und Bilderkennung gesammelt. Ein weiteres zentrales technisches Ergebnis ist die Quantum Machine Learning (QML) Bibliothek „sQUlearn“, die eine systematisierte und modularisierte Implementierung der in AutoQML verwendeten Algorithmen darstellt. sQUlearn ist außerdem als stand-alone Werkzeug für technisches Personal und Wissenschaftler nutzbar. Geordnet nach den Arbeitspaketen des Projekts folgt hier eine detaillierte Auflistung der Ergebnisse.

AP 1: Anwendungsfelder und Anforderungen

Zu Beginn des Projekts wurden die grundlegenden Anforderungen der Anwendungsfälle gemeinsam mit den Industriepartnern erarbeitet und erörtert. Hierbei ging es darum, die eigentliche industrielle Fragestellung mit den Randbedingungen des Projekts in Einklang zu bringen. Am Beispiel der vier untersuchten Use Cases wurden wesentliche Erkenntnisse über die Herausforderungen gewonnen, die auftreten, wenn reale Anwendungen auf (NISQ-)Quantencomputern bearbeitet werden. Hinzu kamen die Anforderungen durch das noch zu erweiternde Automatisierungsframework Auto(Q)ML.

In Bezug auf die Eignung für Quantencomputer wurden verschiedene Methoden der Dimensionsreduktion untersucht, um das jeweilige Problem den aktuell noch bestehenden Beschränkungen der Quantenhardware anzupassen.

In anderen Fällen (z.B. in den Anwendungsfällen IAV und KEB) musste aber auch auf synthetische Daten zurückgegriffen werden. Bei der Auswahl der Alternativdatensätze wurde darauf geachtet, dass die Struktur des Problems so nah wie möglich am ursprünglichen Problem ist, so dass keine Änderung der Fragestellung notwendig ist und ggf. bei Fortschritt der Hardware reibungslos zur Originalproblemstellung zurückgekehrt werden kann.

So stellte sich der Anwendungsfall der IAV zu Beginn des Projekts als zu hochdimensional und komplex für erste QML-Versuche heraus. Daher wurde die Methodenentwicklung zunächst an einem öffentlich verfügbaren Datensatz durchgeführt. Im späteren Verlauf des Projekts wurden die entwickelten Methoden dann auf einen realen Datensatz übertragen.

Beim Anwendungsfall von KEB brachte die Anforderungsanalyse hervor, dass der bereitgestellte Datensatz zu wenige fehlerhafte Zustände aufweist. Da das eigentliche Ziel des Anwendungsfalls darin besteht, Fehlerzustände zu erkennen, sind die Daten für die geplanten supervised-learning Methoden nicht geeignet gewesen. Daher wurde auf einen synthetischen Datensatz ausgewichen. Der neue Datensatz simuliert Signale eines Vibrationssensors gehört zur gleichen Problemklasse, wie der ursprüngliche Datensatz (Klassifikation von Zeitreihen), so dass eine Übertragbarkeit der Ergebnisse gewährleistet wurde.

Detaillierte Ausführungen zu den obigen Punkten sind in der Dokumentation der Arbeiten in den Arbeitspaketen 5 und 6 zu finden.

Meilensteine für AP 1

Für jeden Anwendungsfall wurde ein Anforderungskatalog erstellt.

Im Rahmen des Meilensteins M1 wurden die zu verwendenden Daten bestimmt, aufbereitet und die mathematische Formulierung der Anwendungsfälle fixiert. **Der Meilenstein wurde daher vollständig und fristgerecht erreicht.**

AP 2: Methodik für automatisiertes Machine Learning mit Quantencomputern

Analyse von Encoding Circuits

Zu Beginn des Projektes AutoQML wurde die Kodierung klassischer Daten in Quantenzustände mithilfe von Encoding Circuits, untersucht. Diese sind entscheidend für das Quanten Machine Learning und werden in Quanten-Kernelmethoden sowie Quanten-Neuronalen Netzen (QNN) eingesetzt. Zu Beginn wurden verschiedene Encoding Circuits aus der Literatur implementiert und hinsichtlich ihrer Eignung verglichen. Dabei stellte sich heraus, dass unterschiedliche Circuits für verschiedene Anwendungen geeignet sind. Außerdem wurden verschiedene Enkodierungstechniken, sowohl mit als auch ohne trainierbare Parameter, betrachtet, und die Expressivität der unterschiedlichen Circuits analysiert. Auf Basis dieser Erkenntnisse wurden eigene Encoding Circuits entwickelt und in sQUlearn (siehe AP 4) verfügbar gemacht. In sQUlearn ist es auch möglich, verschiedene Encoding Circuits miteinander zu kombinieren oder eigene Circuits mit einer vorgegebenen schichtweise Struktur auf Basis von Strings zu erstellen. Die automatisierte Auswahl der geeigneten Encoding Circuits im AutoQML-Framework erfolgt als Hyperparameter, der von AutoQML optimiert wird. Zusätzlich wurde ein Werkzeugkasten zur Analyse von Encoding Circuits entwickelt, der die Quanten Fisher Informationsmatrix nutzt. Diese Matrix zeigt, wie sich ein Quantenzustand in Abhängigkeit von den Parametern des Encoding Circuits verändert. Dadurch können redundante Parameter identifiziert und entfernt werden, was die Expressivität der QML-Modelle verbessert. Die entwickelten Werkzeuge wurden erfolgreich in das QML-Framework sQUlearn integriert.

Aufbau von HPC Ressourcen für die Simulation von Quantenschaltkreisen

Im Rahmen des Projekts AutoQML wurden im Bereich Hardware umfassende Maßnahmen zur Einrichtung eines GPU-Hochleistungsrechners als Simulator für Quantencomputer ergriffen. Am Fraunhofer IAO wurde ein abgesichertes Rechnernetz für das GPU HPC-Cluster etabliert, welches eine IT-sicherheitstechnische Isolation von anderen IT-Systemen am Institut gewährleistet. Diese Maßnahme ermöglicht einen sicheren Zugriff auf das HPC-System sowohl intern als auch öffentlich.

Für die Nutzung des AutoQML-Frameworks sowie die Entwicklung der Algorithmen standen in diesem HPC-Cluster zwei Managementserver (Controller) und drei Supermicro Hochleistungsrechner (Nodes) zur Verfügung. Die Managementserver sind verantwortlich für die Benutzer- und Rechteverwaltung sowie das Management der auszuführenden Rechenaufgaben (Jobs). Die Cluster-Management-Software "Slurm" sorgt für eine effiziente Lastverteilung (Load Balancing) der Jobs auf die drei Hochleistungsrechner.

Zusammenfassend wird das HPC-System im Rahmen des Projekts AutoQML intensiv für die Simulation von Quantenschaltkreisen genutzt, was zu einer drastischen Reduzierung der Berechnungszeiten für Anwendungen des AutoQML-Frameworks führt. Diese Optimierungen ermöglichen die Simulation komplexer Quantenschaltkreise zur Weiterentwicklung und zum Training der (Q)ML-Modelle im Projekt. Zudem wurden die HPC-Systeme für die Entwicklung und das Benchmarking des AutoQML-Frameworks eingesetzt.

Datenvorbereitung auf dem Quantencomputer

Im Rahmen der Datenvorverarbeitung wurde die Feature Auswahl mithilfe von Quanten-Annealing auf D-Wave Quantencomputern untersucht. Die Feature Auswahl ist ein essenzieller Bestandteil jeder QML Pipeline, da sie die relevantesten Merkmale aus einem Datensatz auswählt wird. Dies ist insbesondere im Quantencomputing wichtig, da die Zahl der nutzbaren Dimensionen hardwarelimitiert ist. Die Mutual Information (MI) dient hierbei als Maß zur Bewertung der Relevanz eines Merkmals in Bezug auf die Zielvariable. Merkmale mit hoher MI sind stark mit der Zielvariable korreliert und sollten daher im ML-Modell berücksichtigt werden. MIQUBOs (Mutual Information Quadratic Unconstrained Binary Optimization) repräsentieren Optimierungsprobleme, die sich besonders gut auf Quantencomputern lösen lassen. D-Wave Systeme sind speziell dafür konzipiert, solche QUBOs effizient zu verarbeiten. Im Rahmen von AP 2 wurde die Feature Selection mit dieser Methode erfolgreich durchgeführt, indem das Optimierungsproblem mithilfe von MIQUBOs formuliert und auf der Quantenhardware von D-Wave Systemen gelöst wurde. Die Ergebnisse der Feature Selection für den Zeppelin Use-Case sind in Abbildung 1 dargestellt. Ein ML-Modell mit den reduzierten Features zeigt eine signifikante Verbesserung der Leistung. Die Resultate wurden veröffentlicht auf dem 33. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning präsentiert und veröffentlicht [1]. Weiterhin wurde

eine Arbeit zur Erklärung von Klassifikationsdatensätzen erstellt und als pre-print veröffentlicht [13]. Das Problem wurde als QUBO umformuliert und exemplarisch mit Quantencomputersimulatoren gelöst.

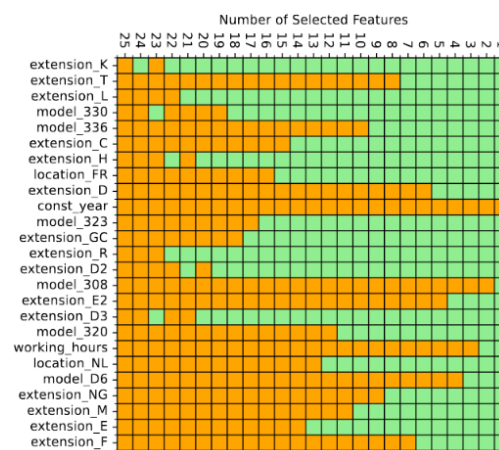


Abbildung 1: Ergebnisse der Feature Auswahl für den Zeppelin Use Case basierend auf MI. Das klassisch harte Optimierungsproblem wurde mittels echten Quantenhardware von D-Wave in einem hybriden Verfahren effizient gelöst. Das Schaubild zeigt für eine bestimmte Anzahl von ausgewählten Features die Kombination, die die Conditional MI maximiert.

Verbesserung der Robustheit von Quanten Neuronalen Netzwerken

Als Indikator für die Robustheit und Generalisierbarkeit von QML Modellen, insbesondere auf Ebene der Quantenschaltkreise, wurden Trainingsverfahren aus dem klassischen maschinellen Lernen, die auf Lipschitz-Konstanten basieren, übertragen. Die Lipschitz-Konstante wird mithilfe der Gewichte des Quantenschaltkreises bestimmt und minimiert, indem ein zusätzlicher Lipschitz-Term als Regularisierung in das Optimierungsproblem aufgenommen wird. Es wurde gezeigt, dass Quantenlernmodelle dadurch robuster gegenüber kohärenten Kontrollfehlern auf der Hardware und gegen Rauschen in den Daten werden. Im Experiment wurde zusätzlich demonstriert, dass eine kleine Regularisierungskopplung die Generalisierungseigenschaften des QML Modells verbessert. Ein weiteres Ergebnis aus diesem Experiment ergab sich aus dem Vergleich zwischen QNNs mit und ohne parametrisierte, trainierbare Encodings. Es stellte sich heraus, dass ohne trainierbares Encoding, wie es häufig in der Literatur durchgeführt wird, die Robustheitseigenschaften des QML Modells nicht über Lipschitz-Regularisierung verbessert werden können. Die wissenschaftliche Arbeit entstand in Kooperation mit der Universität Stuttgart. Die Ergebnisse wurden bei der größten QML-Konferenz, der QTML 2023 am CERN, präsentiert und in dem Journal Physical Review Research veröffentlicht [2].

Entscheidungsmethoden zwischen konventionellen und Quanten Machine Learning Methoden

Ein weiterer Forschungsbereich in AP 2 war der Vergleich zwischen klassischen und quantenbasierten ML-Algorithmen sowie deren automatisierte Auswahl. Dazu wurden zunächst Ansätze untersucht, die einen Vergleich zwischen klassischen und Quanten-Methoden ermöglichen. Für Kernel-Methoden wurde die „Geometric Difference“ analysiert, die einen a-priori Vergleich zwischen klassischen und Quanten-Kernels ermöglicht. Für QNNs wurde ein Ansatz aus der Literatur betrachtet, der die klassische Fisher-Informationsmatrix für den Vergleich von QNNs und klassischen Neuronalen Netzen nutzt. In der Praxis erwies sich eine a-priori Auswahl anhand dieser Kriterien leider als nicht realistisch umsetzbar. Beide Kriterien erfordern aufwändigere Berechnungen als das direkte Training der Modelle, wodurch es deutlich effizienter ist, verschiedene Modelle zu trainieren und zu evaluieren, um anschließend a posteriori das beste Modell auszuwählen. Basierend auf diesen Ergebnissen wurde die automatisierte Auswahl zwischen klassischen und quantenbasierten ML-Methoden durch die „Combined Algorithm Selection and Hyperparameter Optimization“ (CASH) innerhalb des AutoQML-Frameworks gelöst. In der CASH-Optimierung werden die Modellauswahl und die zugehörigen Hyperparameter gemeinsam optimiert, wie dies auch in bestehenden AutoML-Frameworks realisiert wird. Die Auswahl und zeitgleiche Optimierung der Methoden wurde hierbei durch Optuna umgesetzt.

Meilensteine für AP 2

Für Meilenstein M2 wurde innerhalb von AP2 ein Anforderungskatalog für die QML-Realisierungen der verschiedenen Use Cases des Projektes erstellt. Dieser Katalog umfasst die Auswahl der möglichen QML-Algorithmen, die Herausforderungen bei der Umsetzung auf dem Quantencomputer und die potenziellen Anpassungen, die für eine QC-Lösung bei den Use Cases erwartet werden.

Im Jahr 2023 wurde im Rahmen von Meilenstein M3 eine Studie zur Methodik des AutoQML-Frameworks verfasst und veröffentlicht [3]. Die Studie erläutert die Auswahlprozedur des zugrundeliegenden klassischen AutoML-Frameworks Ray sowie die Konzeption des im Projekt entwickelten Auto-QML-Frameworks. Sie diskutiert außerdem die Integration von QML-Algorithmen in AutoML-Prozesse mithilfe des im Projekt entwickelten Pythonpakets sQULearn sowie die Integration des Frameworks in die Plattform PlanQK. **Der Meilenstein für AP 2 wurde daher vollständig und fristgerecht erreicht.**

AP 3: Algorithmen Quantum Machine Learning

Das Gesamtziel von AP 3 besteht aus der Untersuchung, Implementierung, Selektierung und Benchmarking von QML-Algorithmen. Hierzu fand während des Berichtszeitraums neben einer umfassenden Untersuchung zum Stand der Technik auch eine Auswahl der vielversprechendsten QML-Algorithmen im Hinblick auf die vier anwendungsbezogenen Use Cases des geplanten AutoQML-Frameworks statt. Die State-of-the-Art-Analyse umfasst dabei sowohl QML-Ansätze als auch verschiedene QC-Algorithmen und Gradientenverfahren, sowie QNNs, Quantum Support Vector Machines (QSVM), Quantum Genetic Algorithms (QGA), quanteninspirierter Random Forests bestehend aus Quantum Decision Trees, Variational Quantum Eigensolver (VQE), Quantum Approximate Optimization Algorithm (QAOA), Harrow-Hassidim-Lloyd (HHL), Quantum Natural Gradients (QNG), Quantum Natural Simultaneous Perturbation Stochastic Approximation (QNPSA).

Aus der entstandenen Analyse des Unterarbeitspakets 3.1 »Untersuchung und Prototyping bestehender QML-Algorithmen« haben sich grundsätzlich zwei QML-Ansätze herauskristallisiert, die näher untersucht wurden: Die QNNs und Quantum Kernel Methods (QKM). Sowohl QNNs als auch die QKM sind vielseitig einsetzbar, weshalb sie für das geplante AutoQML-Framework genutzt werden sollen. Beispielsweise können diese QML-Ansätze für Hauptkomponentenanalysen, Klassifikationen, Regressionen und Bildverarbeitungsprozesse genutzt werden.

Im Rahmen der Unterarbeitspakete 3.1 sowie 3.2 fand ebenfalls die Implementierung von QML-Algorithmen sowohl auf QC-Simulatoren als auch prototypisch auf realen Quantencomputern statt. Hierfür wurden als Grundlage für die Implementierungen die Erkenntnisse aus der angefertigten Analyse zum Stand der Technik verwendet. Für QNNs wurde ein eigenes Framework entwickelt, das sowohl ein effizientes Training als auch eine Auswertung von QNNs erlaubt. Das Modul erlaubt eine gradientenbasierte Optimierung von QNNs mit mehreren Eingangs- und Ausgangsgrößen und beliebigen Feature-Maps. Die aktuellen Werkzeuge von IBM für die Fehlermitigation werden dabei vollständig unterstützt. Dieses Framework wurde in AP Grundlage der QNN Engine in sQULearn verwendet und paketierte. Im Jahr 2023 wurden die implementierten Algorithmen probeweise auf einem Ionenfallen basierten Backend (IonQ) getestet und mit den Ergebnissen auf einem Supraleiter basierten Backend (IBM Quantum System One) verglichen.

Planmäßig hätte Arbeitspaket 3 vollends im ersten Projektjahr bearbeitet werden sollten. Die entsprechenden Meilensteine M1 und M2 wurden ebenfalls gehalten bzw. geliefert (siehe Gantt-Chart und Zwischenbericht 2022). Wie in Zwischenbericht 2022 allerdings angemerkt wurde, war Arbeitspaket 3 von einer leichten Umpriorisierung der Projektaufgaben betroffen, um genau diese zu halten. Es wurden hauptsächlich nur Algorithmen, die für die direkte Umsetzung der Use Cases relevant sind, implementiert. Weitere Methoden zur alternativen Umsetzung oder der Umsetzung neuer Anwendungsfälle mittels des Frameworks wurden nicht im Fokus behandelt.

Dies wurde 2023 nachgeholt. Einerseits stand die Implementierung von Trainable-Encoding-Algorithmen mit Lipschitz-Regularisierung im Fokus der Arbeiten. Ein zentrales Ergebnis dieser Untersuchung, die in Kollaboration mit dem ICP und dem IST der Universität Stuttgart entstand, ist u. a., dass mithilfe eines trainierbaren Encodings einige Verbesserungen gegenüber in der Literatur üblichen Quanten Neuronale Netzen zu finden sind. Jedes Quanten Neuronale Netz kann alternativ als partielle Fourierreihe dargestellt werden. In den typischen Data-Reuploading Modellen mit fester Datenenkodierung werden jedoch lediglich die Amplitude der Fourierkoeffizienten trainiert und nicht auch deren Frequenzen. Durch trainierbare Enkodierung sind nicht nur die Amplituden, sondern auch die Frequenzen der partiellen Fourierreihe trainierbar und führen zu flexibleren und mächtigeren Quanten Lernmodellen. Andererseits wurden Quantengaußprozesse detailliert beleuchtet und modular implementiert. Eine Anwendung fanden diese Algorithmen in der quantengestützten Hyperparameteroptimierung in AP 4.3.

Zuletzt wurden die Benchmarks der Anwendungsalgorithmen abgeschlossen und die jeweils passenden zur Bearbeitung der Use Cases in AP5 und AP6 ausgewählt (siehe Ausführungen zu AP5 und 6).

Meilensteine für AP 3

Alle Meilensteine wurden fristgerecht und umfassend erreicht.

AP 4: AutoQML Developer-Suite: Werkzeuge und Komponenten

Schnittstellen und Einordnung in das AutoML-Framework

Im Projekt wurden zunächst 45 verfügbare AutoML-Frameworks analysiert und hinsichtlich mehrerer (harter und weicher) Kriterien bewertet. Frameworks, welche den Anforderungen nicht genügten, wurden verworfen. Die verbleibenden 8 Frameworks wurden nach Abstraktionsebene eingruppiert und es wurden für jedes Framework ein Minimal Working Example erstellt und allen Partnern zur Verfügung gestellt. Die Use Case-spezifischen Anforderungen an das AutoML-Framework wurden anhand von standardisierten Fragebögen und Experten Interviews erarbeitet.

Werkzeuge und Komponenten

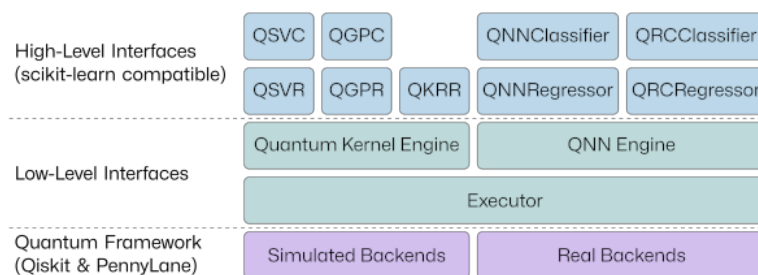


Abbildung 2: Schematischer Aufbau von sQLearn. Die QNN Engine wurde zum Teil in AP 3 und im Projekt H2Giga Degrad-EL3-Q¹⁸ erarbeitet. Die Quantum Kernel Engine wurde auf Basis einer modularen Quantenkernelimplementierung, die im Projekt AQUAS (gleiche Fördertranche) erfolgte abstrahiert und in sQLearn paketiert. Die restlichen Komponenten wurden im Rahmen des Projekts AutoQML entwickelt.

Im Jahr 2023 wurde das open-source Python Framework sQLearn entwickelt, um QML-Methoden nahtlos in bestehende Machine-Learning-Frameworks zu integrieren. Es bietet eine scikit-learn Schnittstelle und umfasst high- und low-level Methoden (siehe Abbildung 2). High-level Methoden erleichtern die Nutzung und Integration für unerfahrene Benutzer, während low-level Methoden für Experten zur Entwicklung eigener Experimente geeignet sind. sQLearn unterstützt QNN, QKMs und Quantum Reservoir Computing (QRC), und bietet Features wie die Berechnung von beliebigen Ableitungen auf dem Quantencomputer. Der im Framework enthaltene Executor verbessert die Benutzererfahrung durch automatische Session-Verwaltung und Fehlerbehandlung und unterstützt die einfache Einbindung von Simulatoren und Quantencomputern in den Frameworks Qiskit und PennyLane. sQLearn ist Open-Source und auf GitHub verfügbar¹⁹, mit ausführlicher Dokumentation und Beispielen. Das Softwarepaket wird aktiv betreut und stetig weiterentwickelt. Eine Publikation mit dem Titel “sQLearn: A Python Library for Quantum Machine Learning” wurde im Journal “IEEE Software” zur Veröffentlichung akzeptiert [4]. Die QNN Engine basiert auf Arbeiten des AP 3 und des Projekts H2Giga DEGRAD-EL3-Q, die Quantum Kernel Engine basiert auf modularen Implementierungen, die im Projekt AQUAS²⁰ entstanden sind. Die restlichen Komponenten, die Vereinheitlichung, die Modularisierung und die Aufbereitung als Open-Source Paket (CI/CD Pipeline, Testing etc.) erfolgte in AP 4 dieses Projekts. Die flexible Paketierung bietet den Grundstein für die Anbindung an die AutoML Frameworks.

Unterauftrag HQS

Siehe den Punkt „Zusammenarbeit mit anderen Stellen“ auf Seite 7.

QC-optimierte Algorithmenauswahl und Hyperparameteroptimierung für AutoQML

Eine Hyperparameteroptimierung basierend auf Quanten-Gaussprozessen (QGP) wurde entwickelt und getestet. QGPs orientieren sich an klassischen Gaußprozessen und nutzen »Fidelity-Typ Kernel« zur Bildung der Kovarianzmatrizen. Das Verfahren wurde für Regressionsprobleme auf QC-Simulatoren und echten Quantencomputern angewendet. Der Einfluss von Feature-Map Variationen und Datenskalierung auf die Lösungsqualität wurde

¹⁸ <https://www.ipa.fraunhofer.de/de/referenzprojekte/Degrad-EL3-Q.html>

¹⁹ <https://github.com/sQLearn/sqlearn>

²⁰ https://www.digitale-technologien.de/DT/Redaktion/DE/Standardartikel/Quanten-Computing/qc_projekt_aquas.html

analysiert und Regularisierungstechniken wurden eingesetzt, um mathematische Anforderungen unter »Shot-Noise« zu erfüllen. Die Parameter des Quantenschaltkreises wurden anhand der »Log-Likelihood Kostenfunktion« optimiert. QGP wurde als Surrogat-Modell in die »Bayes'sche Optimierung« (BO) integriert, was zur Entwicklung der Quanten Bayes'schen Optimierung (QBO) führte. Das Verfahren wurde mit dem »Zeppelin« Datensatz getestet und die Ergebnisse mit klassischen Algorithmen verglichen. QGPs und QBO sind in sQUlearn implementiert, und ein Tutorial dazu ist in der sQUlearn Dokumentation verfügbar. Die Ergebnisse wurden in Quantum Machine Intelligence veröffentlicht [5]. Die QBO war wurde im Rahmen einer Masterarbeit erarbeitet [14].

Durchgängige und dynamische Werkzeugketten

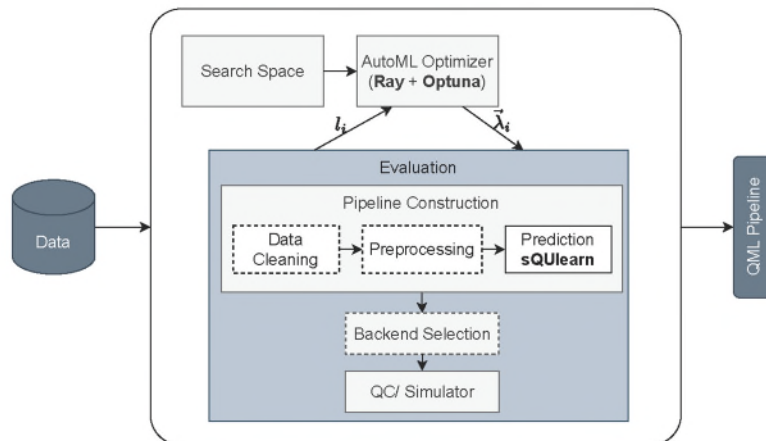


Abbildung 3 Architektur Schema-Darstellung des AutoQML Frameworks

Ein Whitepaper zur Auswahl geeigneter open-source AutoML Frameworks für die Integration von QML-Ansätzen wurde veröffentlicht [3]. Es bietet eine Marktübersicht und analysiert Frameworks anhand von Eigenschaften wie Programmiersprachen, Community-Stärke und unterstützten ML-Backends. Die Frameworks wurden in high- und low-level Klassen unterteilt und bewertet, wobei Ray und AutoGluon als die besten identifiziert wurden. Ray wurde für das AutoQML-Framework gewählt, da es die Implementierung aller Use-Cases und die Integration von sQUlearn ermöglicht. Das AutoQML-Framework automatisiert typische ML-Pipelines und QC-spezifische Aspekte und verwendet Ray für die Hyperparameteroptimierung. Es bleibt kompatibel mit scikit-learn und ist an das PlanQK Ökosystem angebunden. Zwei Whitepapers sind über arXiv und die Projektwebsite verfügbar [3,6]. Intensive Workshops und Beratungen unterstützten die Entwicklung des AutoQML-Frameworks.

Um die QML-Methoden von sQUlearn in das AutoQML Framework zu integrieren, wurden Wrapper-Klassen erstellt, die die Methoden für Klassifikation und Regression einbanden. Diese Klassen implementierten Funktionen zur Initialisierung, zum Training und zur Vorhersage. Die Trainingsfunktion baute die sQUlearn Klasse mit vordefinierten Hyperparametern auf, einschließlich der Kodierungsschaltkreise und Kernel-Objekte. Die Dimension der Schaltkreise wurde durch die Anzahl der Qubits und die Wiederholungen der Feature-Kodierungen gesteuert. Weitere Hyperparameter konnten ebenfalls übergeben werden. Die Vorhersagefunktion entsprach derjenigen in sQUlearn. Die `sample\_configuration` Funktion erstellte Konfigurationen basierend auf Hyperparametern und Einschränkungen. Ein sQUlearn Executor Objekt wurde verwendet, um die Ausführung der Quantenschaltkreise zentral zu verwalten, sodass keine zusätzlichen Implementierungen im AutoQML Framework erforderlich waren. Das AutoQML Framework, welches zunächst als privates Github-Repository umgesetzt, wurde zum Ende des Projekts der Öffentlichkeit als Open-Source-Plattform zugänglich gemacht. Der Projektpartner GFT integrierte das AutoQML Framework zudem in das PlanQK-Ökosystem.

Das finale Framework ist auf GitHub verfügbar.²¹

²¹ <https://github.com/AutoQML/autoqml>

Meilenstein für AP 4

Die Entwicklung von sQUlearn, was eine Integration der Algorithmen und eine Cloudanbindung an verschiedene Hardwareprovider bietet, beschließt M2. Die AutoML-Hyperparameteroptimierung und kombinatorische Algorithmenauswahl mittels Quantencomputing (M3) wurde mit der in [1] entwickelten Methode erforscht, implementiert und auf den im Projekt laufenden Anwendungsfall der Zeppelin GmbH angewendet. **Damit ist der Meilenstein fristgerecht und vollständig erreicht worden.**

AP 5: Applikationen: Produktion

Anwendungsfall: Bilderkennung in der lasergestützten Fertigung (TRUMPF)

Der Anwendungsfall wurde vom Projektpartner Trumpf zur Verfügung gestellt und maßgeblich vom IAO gelöst. Die Problemstellung besteht in der Erkennung von metallischen Stützlamellen beim Laserschneiden. Zunächst wurden gemäß AP1 die Daten und die Problemstellung fixiert. In einem ersten Schritt wurden dann neben dem eigentlichen Datensatz (TRUMPF) die Methoden auch an zwei Benchmark Datensätzen zur Bildverarbeitung erprobt (MNIST und fashionMNIST).

Dimensionsreduktion

Um den Anwendungsfall für NISQ-Quantenalgorithmen lösbar zu machen, wurden verschiedene Methoden zur Dimensionsreduktion angewandt, darunter PCA, t-SNE, dichte Autoencoder und Faltungsautoencoder. Basierend auf Techniken für große Sprachmodelle wurden auch Vision-Transformer-basierte Architekturen implementiert. Unter den oben genannten erwiesen sich Faltungs-Autoencoder und Vision-Transformer-basierte Architekturen als am effizientesten für die Konstruktion von Quantenkernen mit kleinen Datensätzen auf GPUs.

QML-Methoden

Die Anwendung von Quantenkernel-Methoden auf den Anwendungsfall durch die Implementierung von QSVM wurde erforscht. Es wurden auch verschiedene Quanten Feature Maps untersucht. Die Erhöhung der Anzahl der Qubits verbesserte die QSVM-Leistung auf dem TRUMPF-Datensatz, aber nicht auf den standardisierten Benchmark-Datensätzen. Projizierte Quantenkernel schnitten - bezogen auf die Accuracy - durchweg besser ab als Quantenkernel (vgl. Abbildung 4). Darüber hinaus wiesen projizierte Quantenkerne eine ähnliche oder manchmal sogar bessere Performanz auf als klassische Kernel. Die Optimierungsmethode »Kernel Target Alignment« führte nicht zu signifikanten Verbesserungen gegenüber dem Standardansatz. Neben Simulationen wurden auch Experimente auf realer QC-Hardware von IBM für kleinere Teile der Datensätze erfolgreich durchgeführt. Diese Forschung zu QSVM-Anwendungen führte zu der Masterarbeit [7] an der Universität Stuttgart.

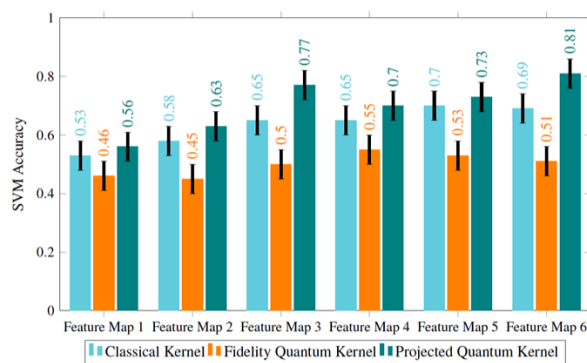


Abbildung 4 Vergleich von Genauigkeit der unterschiedlichen Kerntypen bei Verwendung eines Convolutional Autoencoders zur Dimensionalitätsreduktion auf dem TRUMPF Datensatz.

Es wurden auch quantenklassische Convolutional Neural Networks (QCCNNs) entwickelt und ihre Leistung mit vergleichbaren klassischen CNNs verglichen. Nicht trainierbare QCCNNs mit zufällig generierten Quantenschaltkreisen zeigten auf allen getesteten Datensätzen eine ähnliche Genauigkeit wie klassische CNNs und übertrafen sogar das klassische Modell auf einem TRUMPF-Datensatz. Erste Ergebnisse für trainierbare QCCNNs zeigten eine Leistung auf Augenhöhe mit der nicht trainierbaren Variante. Ein längeres Training war jedoch aufgrund des Rechenaufwands für die erneute Ausführung der Quantenschaltkreise bei jeder Iteration nicht möglich. Darüber hinaus schneidet ein QCCNN mit zufällig generierten, nicht trainierbaren Quantenschaltkreisen mit Messungen in der Mitte des Schaltkreises tendenziell schlechter ab als das klassische CNN.

Die Anforderung, während des QCCNN-Trainings zahlreiche Quantenschaltkreise auszuführen (ca. 4000 pro Bild und Iteration für den TRUMPF-Datensatz), stellt sowohl die Simulation als auch die reale Hardware vor große Herausforderungen. Um diesen Rechenanforderungen gerecht zu werden, haben wir kleinere Bilder aus dem synthetischen Datensatz von TRUMPF erstellt, um den Trainingsprozess überschaubarer zu machen. Die Ergebnisse wurden in [8] veröffentlicht.

Trotz allem erwies es sich als schwierig, den gesamten Trainingsprozess für ein Sample von 400 Bildern auf Hardware durchzuführen. Die folgenden Resultate in Abbildungen 5 und 6 basieren daher auf einem mittels Simulationen trainiertem Netzwerk, bei welchem nur die Inferenz über Rechnungen auf echter Hardware (auf dem Backend IBM Torino) erfolgte. Hierbei ergab sich eine Reduktion der Genauigkeit von etwa 95% auf 85% durch Verwendung von echter Hardware.

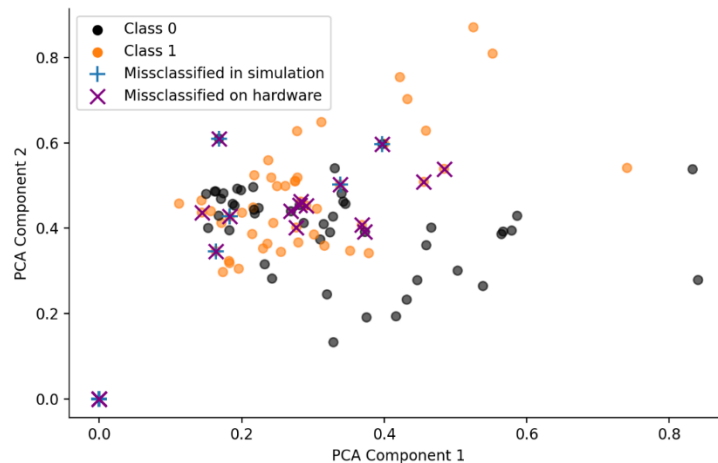


Abbildung 5: Vergleich der Vorhersagegenauigkeit zwischen Simulation und Hardwareergebnissen. Bei der Inferenz auf echter Hardware unterlaufen dem Klassifikator erwartungsgemäß mehr Fehler. Um die vieldimensionalen Datenpunkte in 2D repräsentieren zu können, wurden sie nur über ihre ersten beiden Hauptkomponenten dargestellt. Die Klasse 1 entspricht einer vorhandenen Stützlamelle, die Klasse 0 einem leeren Sockel.

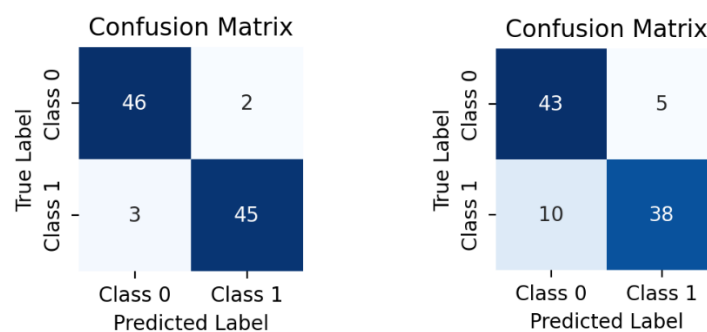


Abbildung 6: Konfusionsmatrizen zu Abbildung 5. Links mit Inferenz über Simulation. 5 Fehler entsprechen 94,8% Genauigkeit. Rechts mit Inferenz über Hardwarerechnung. 15 Fehler entsprechen 84,4% Genauigkeit.

Anwendungsfall: Betriebszustände autonomer Fahrzeuge

Der Anwendungsfall wurde vom Projektpartner KEB bereitgestellt und maßgeblich von den Partnern USU und IPA gelöst. Der Anwendungsfall ist die Klassifikation von Betriebszuständen von autonomen Fahrzeugen in der Produktion. Die Fahrzeuge können verschiedene (Fehler-)Zustände aufweisen, deren frühzeitige Erkennung auf Basis von Sensordaten eine Vermeidung von Ausfällen erlauben würde.

Im ersten Projektjahr wurde der Anwendungsfall genau spezifiziert, Daten analysiert und eine erste Lösung des Anwendungsfalls mit klassischen Machine Learning Methoden durch die USU erarbeitet. Hierbei hat sich jedoch herausgestellt, dass der bereitgestellte Datensatz zu wenige fehlerhafte Zustände aufweist, so dass kein zufriedenstellendes Training durchgeführt konnte. Die Partner KEB, USU und Fraunhofer IPA einigten sich daher darauf,

zusammen einen synthetischen Datensatz (Vibrationsdaten) zu identifizieren, der anstelle des ursprünglich geplanten Datensatzes zum Einsatz kam. Der neue Datensatz gehört zur gleichen Problemklasse, wie der ursprüngliche Datensatz (Klassifikation von Zeitreihen), so dass eine Übertragbarkeit der Ergebnisse gewährleistet ist. Für die Lösung des Anwendungsfalls mit QML wurden verschiedene Strategien für die Verarbeitung des Problems auf dem Quantencomputer erarbeitet. Die finale Lösung basiert auf der Überführung der Zeitreihendaten in Spektrogramme (siehe Abb. 7).

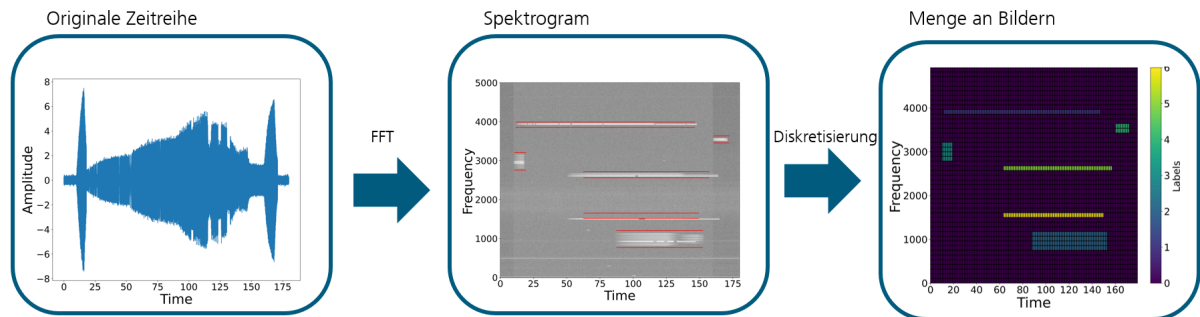


Abbildung 7: Umwandlung der generischen Zeitreihendaten in Bilddaten mittels Fourier Transformation

Das Spektrogramm wurde in kleine Teilbereiche unterteilt, so dass der finale Datensatz aus 3.291 kleinen Bildausschnitten der Größe 7x30 Pixeln besteht. Jedem Bildausschnitt kann eindeutig einer von sieben Betriebszuständen zugeordnet werden. Das Ursprungsproblem wurde somit auf eine Mehrfachklassen Bildklassifikation reduziert. Dieses wurde mit verschiedenen QML Methoden gelöst, wobei der Schwerpunkt auf Quantenkernelmethode (insb. QSVC mit diversen Konfigurationen) lag. Es wurde eine umfangreiche Hyperparameteroptimierung durchgeführt, die sowohl klassische, als auch quantencomputing Parameter optimiert hat. So wurden verschiedene Feature Maps mit verschiedenen Konfigurationen was die Anzahl der Layer und Qubits angeht verglichen. In Simulationen wurden vielversprechende Ergebnisse mit erzielt, deren Performance vergleichbar mit der von klassischen Methoden war.

Das QML Modell (QSVM), das in der Simulation die besten Ergebnisse geliefert hat, wurde für die Ausführung auf IBM Quantencomputern ausgewählt. Hierfür haben wir zunächst Simulationen mit Fehlermodellen genutzt, um die Anwendbarkeit auf der echten Hardware zu evaluieren und die Hyperparameter in Anwesenheit von Rauschen zu optimieren. Anschließend wurde das Modell auf zwei IBM Quantencomputern (IBM Kolkata und IBM Cairo) ausgeführt. Die Ergebnisse waren überraschenderweise von ähnlicher Qualität wie die der fehlerfreien Simulation. Dies lässt sich darauf zurückführen, dass die Kernelmatrix, die die Klassifikationsqualität der Methode bestimmt, strukturell vom Rauschen des Quantencomputers nur wenig beeinflusst ist (siehe Abb. 8). Die einzelnen Elemente sind zwar gedämpft, die Gesamtstruktur konnte allerdings gut erhalten bleiben. Dieser Umstand macht Quantenkernelmethode zu geeigneten Algorithmen für die Ausführung auf echter Quantencomputerhardware.

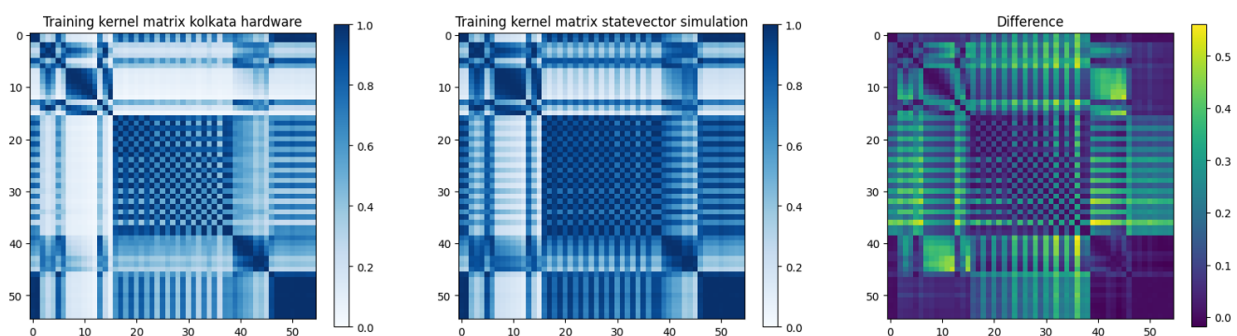


Abbildung 8: Differenz der Einträge in der Quanten-Kernelmatrix zwischen Simulation (Mitte) und realer Hardware (links). Als reales Backend wurde die IBM Kolkata Hardware genutzt.

Meilensteine AP 5

Die Meilensteine beinhalteten die klassische (M1) und quantenmechanische Lösung auf Simulatoren (M3) und (M4). **Die Meilensteine wurden fristgerecht und vollständig erreicht.**

AP 6: Applikationen: Automotive

Anwendungsfall: Bewertung von Gebrauchtmotoren

Der Anwendungsfall wurde vom Projektpartner Zeppelin bereitgestellt und maßgeblich in Zusammenarbeit mit dem IAO gelöst. Es handelt sich um eine Regression zur Findung von Preisen für Gebrauchtmotoren auf der Basis historischer Preise.

Rolle bei der Framework Auswahl

Dieser Anwendungsfall wurde maßgeblich auch dazu genutzt, um das klassische AutoML Framework auszuwählen, auf dessen Grundlage AutoQML errichtet wurde. Dazu wurden in [9] sieben klassische ML-Verfahren gegen drei AutoML Frameworks verglichen.

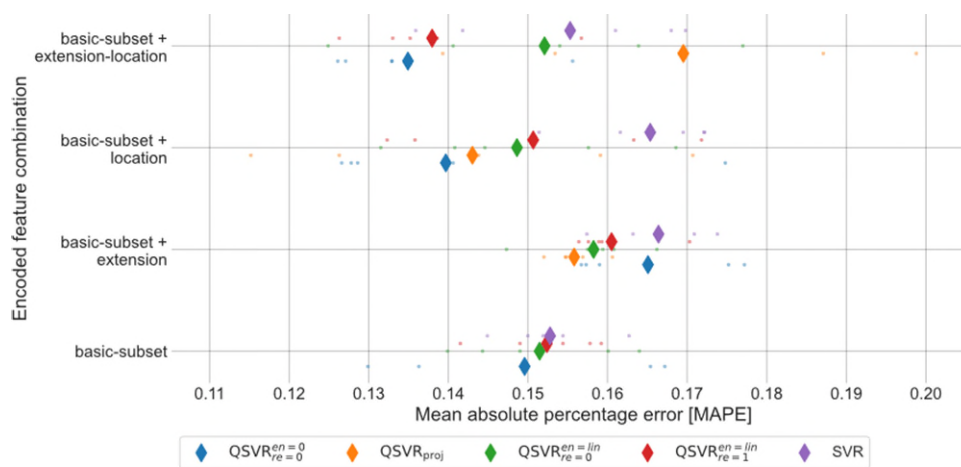


Abbildung 9: Genauigkeit des QSVR Modells auf einem enkodierten Datensatz angegeben in Mean Absolute Percentage Error (MAPE). Jeder Diamant zeigt den Mittelwert der Vorhersage über fünf Wiederholungen. Einzelne Punkte zeigen das Ergebnis eines einzelnen Durchlaufs. Ein Performanzvorteil für QSVRs gegenüber dem auf klassischen radialen Basisfunktions-Kernen basierten Support Vektor Modellen lässt sich identifizieren.

Quantenkernelmethoden

Quantenkernelmethoden wurden als vielversprechende Quantenlösung identifiziert und es wurde ein detaillierter Benchmark zwischen klassischen und quantenbasierten Kernelmethoden durchgeführt. Hierzu wurden sowohl Quantenkernel vom Fidelitäts- als auch Projiziertem-Typ simuliert. Aus den Simulationen geht hervor, dass Quantenkernel von bestimmten Feature Maps existieren, die den klassischen Standard in der Genauigkeit in Bezug auf Mean Absolute Percentage Error übertreffen (s. Abb. 9). Die Ergebnisse dieser Untersuchung sind in [15] niedergelegt.

Feature Selection

Zusätzlich wurde anhand der Zeppelin Daten auch eine Feature Selection über MIQUBOs (Mutual Information Quadratic Unconstrained Binary Optimization) durchgeführt und auf D-Wave Quantenannealern gelöst. Die Ergebnisse in [1] deuten darauf hin, dass der MIQUBO Ansatz in Datensätzen mit geringer MI-Konzentration zu besseren Feature-Kombinationen führen. Weitere Informationen sind im Bericht zu AP2 zu finden

Vergleich mit rbf-Kernel Baseline

Auf einem Teildatensatz (von Caterpillar-308) wurde die Quanten Support Vector Regression simuliert und auf IBM Sherbrooke und IonQ Aria 1 durchgeführt und mit dem Baseline-Modell verglichen (klassischer rbf-Kernel), siehe Abbildung 10.

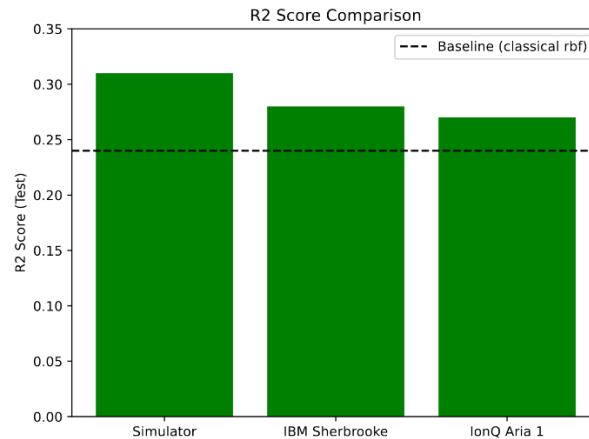


Abbildung 10: Vergleich der resultierenden R2 Test Scores der Quanten Regressions Modelle. Die Quantenmodelle liefern bessere R2 Scores als die klassische Baseline, die auch bei Anwendung von NISQ-Computern besser bleibt.

Anwendungsfall: Zeitreihenvorhersage von Motorfüllzuständen

Der Anwendungsfall wurde von der IAV eingebracht und zunächst mit klassischen Machine-Learning-Methoden bearbeitet. Die Lösung mittels Quantencomputing wurde vom Fraunhofer IPA entwickelt. Es handelt sich um eine Zeitreihenvorhersage im automobilen Kontext. Zu Beginn des Projekts erwies sich der ursprüngliche Anwendungsfall als zu hochdimensional und komplex für erste Versuche mit Quanten-Machine-Learning (QML). Daher wurde die Methodenentwicklung zunächst anhand eines öffentlich verfügbaren Datensatzes zum Internetverkehr durchgeführt. Im späteren Projektverlauf wurden die entwickelten Methoden auf einen von der IAV bereitgestellten Datensatz übertragen, der den Zylinderfüllstand von Verbrennungsmotoren beschreibt.

Bei beiden Datensätzen (Internetverkehr und Zylinderfüllstand) handelt es sich um univariate Zeitreihen, bei denen jeweils der nächste Zeitschritt vorhergesagt werden soll. Dies ist auch bekannt als "one-step ahead forecasting".

Für die Quantenlösung wurden zunächst Quantum Neural Networks (QNNs) eingesetzt. Um das Training auch für größere Datenmengen zu ermöglichen, wurde in sQUlearn Minibatch-Gradient-Descent implementiert. Durch die Einbindung von Ray Tune und die flexible Wahl von Hyperparametern konnten zahlreiche QNN-Konfigurationen effizient trainiert werden. Erste Trainingsergebnisse zeigten, dass QNNs grundsätzlich in der Lage sind, eine One-Step-Ahead-Vorhersage durchzuführen. Die Vorhersage ganzer Zeitreihen mittels QNNs erwies sich jedoch als unzureichend, weshalb alternative QML-Modelle untersucht wurden.

Parallel dazu wurden, unter anderem im Rahmen einer Masterarbeit in Kooperation mit dem Institut für Computerphysik der Universität Stuttgart, Quantenmodelle erforscht, die besser für die Zeitreihenprädiktion geeignet sein sollen [10]. Dabei wurden Quantum Long Short-Term Memory (QLSTM) sowie Quantum Recurrent Neural Networks (QRNNs) untersucht und mit verschiedenen Datensätzen trainiert und analysiert.

Diese Modelle nutzen eine Zellstruktur für jeden Zeitschritt, wodurch vergangene Informationen über längere Zeiträume hinweg gespeichert, aktualisiert und weitergegeben werden können. Sie zeigten eine deutlich bessere Vorhersagequalität als die vorher eingesetzten QNNs, konnten jedoch keine signifikanten Verbesserungen gegenüber klassischen Verfahren erzielen. Die Ergebnisse der Masterarbeit wurden in [11] publiziert und befinden sich derzeit im Begutachtungsprozess.

Ein weiteres zentrales Ergebnis der Masterarbeit war, dass Quantum Reservoir Computing (QRC) deutlich bessere Vorhersagen für Zeitreihen liefern kann als die variationellen Ansätze. Für das finale Modell des IAV-Anwendungsfalls und das Benchmarking (siehe nächstes Kapitel) wurde daher ein QRC-Ansatz gewählt. Dabei handelt es sich um einen randomisierten, komplexen Quantenschaltkreis, der in verschiedenen Basen gemessen wird. Die Messergebnisse dienen anschließend als Eingabe für eine klassische lineare Regression. Diese Methode eignet sich insbesondere auch für die Ausführung auf realen Quantencomputern.

Das Ergebnis der Vorhersage des IAV-Datensatzes mit einem Simulator und mit dem Quantencomputer "ibm_sherbrooke" ist in Abbildung 11 dargestellt.

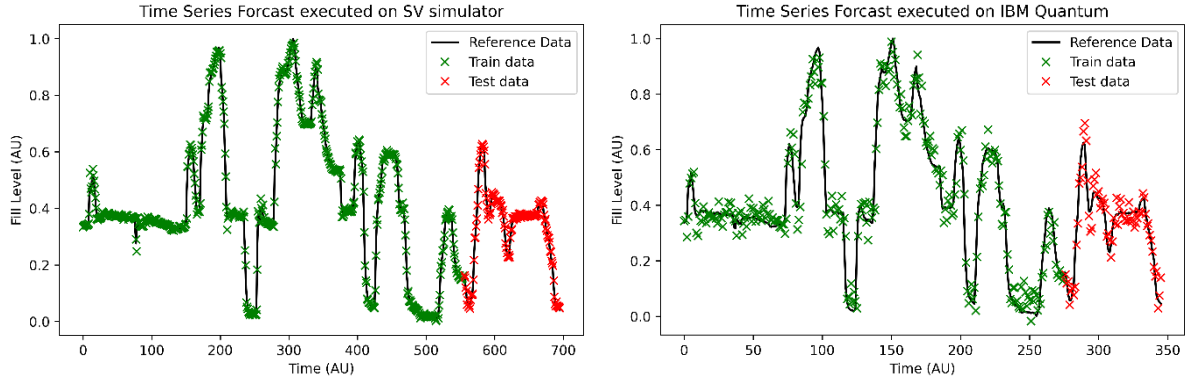


Abbildung 11: One-Step-Ahead-Vorhersage des Füllstanddatensatzes der IAV mit einem Quantum-Reservoir-Computing-Ansatz. Links: Ergebnisse der fehlerfreien Simulation mit dem Statevector-Simulator zeigen eine sehr gute Übereinstimmung mit den Trainings- und Testdaten. Rechts: Ergebnisse desselben Modells, das auf dem IBM-Quantencomputer "ibm_sherbrooke" trainiert und ausgewertet wurde. Die Vorhersage mit dem realen Quantencomputer ist ebenfalls funktionsfähig, jedoch etwas ungenauer als in der Simulation.

Die Meilensteine AP 6

Die Meilensteine beinhalteten die klassische (M1) und quantenmechanische Lösung auf Simulatoren (M3) und (M4). **Die Meilensteine wurden fristgerecht und vollständig erreicht.**

Benchmarking AP5 & AP6

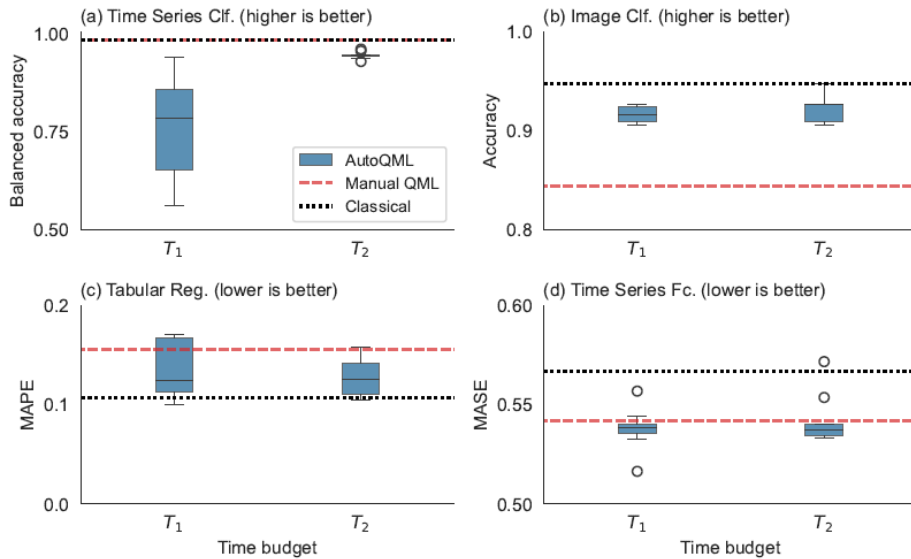


Abbildung 12: Vergleich von AutoQML (boxen, Statistik aus 10 Durchläufen mit klassischen ML Methoden und manuellen QML Pipelines für den (a) KEB, (b) Trumpf, (Zeppelin) und (d) IAV Use Case.

Für den Meilenstein M4 wurde ein umfassendes Benchmarking des entwickelten AutoQML Frameworks auf den vier im Projekt untersuchten Use Cases durchgeführt. Für jeden Anwendungsfall wurden jeweils 10 Rechnungen von AutoQML für ein Zeitbudget von $T_1 = 10.000s$ und $T_2 = 50.000s$ durchgeführt und mit klassischen ML Methoden (GP, KRR, Random Forest, Boosted Trees), sowie den manuellen QML Pipelines aus AP5&6 verglichen. Das Ergebnis ist in Abbildung 12 zu sehen. Die Ergebnisse wurden in einem Artikel veröffentlicht, der nach Projektende auf der Konferenz IEEE World Congress on Services akzeptiert wurde und in den dortigen Proceedings veröffentlicht wird [12].

Fortschreibung des Verwertungsplans.

Das Projekt AutoQML ist die Basis für eine Folgebeauftragung durch das DLR OS im Rahmen des QCI Projekts QCOptSens Cross-Compiler.²² Dort wird AutoQML so weiterentwickelt, dass es zur Bilderkennung von Satellitendaten eingesetzt werden kann. In diesem Rahmen werde die vorhandenen Funktionen zur Bilderkennung durch zusätzliche Innovation erweitert. Das Projekt ist eine Kooperation der Firma HQS Quantum Simulations GmbH mit den Fraunhofer Instituten IAO und IPA. Projektstart war im April 2025 und wird bis Ende 2026 andauern.

Die im Projekt erarbeiteten Methoden fließen in ein Leistungsangebot der Fraunhofer Institute ein. So wird sQUlearn u.a. in einem kostenpflichtigen QML Kurs im Schulungsprogramm der Fraunhofer Academy eingesetzt.²³

sQUlearn wird inzwischen routinemäßig in vielen Projekten der Fraunhofer Institute eingesetzt und trägt aufgrund seiner einfachen Benutzung zu einer maßgeblichen Effizienzsteigerung bei. Das Framework selbst, sowie die eigenständige Entwicklung sQUlearn dienen als Grundlage für ein umfassenderes Toolkit, was im Rahmen des BMWK geförderten Projekts „Full Stack Quantum für Deutschland“ professionalisiert werden könnte.

Aktualisierter Verwertungsplan:

Lfd. Nr.	Bezeichnung	Zeithorizont
1	Präsentation und Bereitstellung der Projektergebnisse durch (peer-reviewed) Veröffentlichungen, Whitepaper und Studien, sowie Konferenzbeiträge durch die Fachabteilungen des Fraunhofer IPA und IAO und die Industriepartner	Ab ca. 12 Monate nach Projektbeginn Bis zu 36 Monate nach Projektende
2	Steigerung der wissenschaftlichen Kompetenz und internationalen Sichtbarkeit der Fraunhofer-Institute in den Bereichen ML und QML (insbesondere automatisiert)	Ab ca. 24 Monate nach Projektbeginn Bis zu 36 Monate nach Projektende
3	Wissenstransfer durch Forschungs- und Innovationsberatung von Industriepartnern aus den Bereichen Produktion und Automotive (perspektivisch auch großer Einfluss auf weitere Branchen im Beratungsportfolio der Fraunhofer-Institute IAO und IPA)	Ab ca. 24 Monate nach Projektbeginn Bis zu 36 Monate nach Projektende
4	Nutzung der Ergebnisse in laufenden Dissertationen als wichtige Basis für die Ausbildung von wissenschaftlichem Nachwuchs	Gesamte Projektlaufzeit Bis zu 36 Monate nach Projektende
5	Integration von AutoML Anwendungen in interne Software der USU	Bis zu 36 Monate nach Projektende
6	Höherer Automatisierungsgrad in IAV ML Pipelines	Bis zu 24 Monate nach Projektende
7	Weiterentwicklung und Evaluierung des AutoML-Konzepts als potenzielles Kernelement für AutoML-gestützte prädiktive Instandhaltungsdienste von KEB	Bis zu 24 Monate nach Projektende
8	Verfeinerung des Algorithmus und Ausdehnung der Anwendbarkeit für das Laserschneiden bei TRUMPF	offen

²² <https://qci.dlr.de/en/hqs-quantum-simulations-supports-qcoptsens-in-the-development-of-a-cross-compiler-for-automatic-sensor-data-analysis-with-qml/>

²³ <https://www.academy.fraunhofer.de/de/weiterbildung/quanten/quantum-machine-learning.html>

9	Preisvorhersage für gebrauchte Baumaschinen in Serientool integriert Erweiterung auf andere Abteilungen und Unternehmensbereiche wie die Neumaschine und Rental bei der Zeppelin GmbH	Im Projektverlauf Bis zu 12 Monate nach Projektende
10	AutoQML auf Planq Website verfügbar gemacht	Zum Projektende

Zu Lfd. Nr. 1 wurden die Arbeiten [1-15] publiziert. Dabei handelt es sich um 8 peer-reviewed Publikationen, eine Submission, ein Konferenzbeitrag, zwei Whitepaper und drei Masterarbeiten. Letztere sind unter Lfd. Nr. 4 noch einmal explizit aufgeführt.

Zu Lfd. Nr. 2 wurde das Projekt und dessen Ergebnisse auf mehreren nationalen und internationalen Veranstaltungen und Konferenzen präsentiert. Über die Projektlaufzeit haben im Rahmen der Öffentlichkeitsarbeit von AP 7 mehr als 50 Veranstaltungen stattgefunden, auf denen AutoQML vorgestellt wurde. Das Projekt hat zu einer großen Sichtbarkeit der beteiligten Fraunhofer Institute beigetragen.

Das Projekt hat durch die außenwirksame Darstellung der Kompetenz im Bereich QML maßgeblich dazu beigetragen, das Fraunhofer Forschungs- und Innovationszentrum 8, gefördert durch die Dieter Schwarz Stiftung einzuwerben.²⁴ Das Zentrum hat eine Laufzeit von 10 Jahren im Schwerpunkt QML und wird von den beiden Instituten IAO und IPA durchgeführt.

Weiterhin wurden mehrere internationale Kooperationen im Bereich QML initiiert, u.a. mit Rigetti Computing Inc. und der University of Waterloo. Mit der University of Waterloo veranstalten die Fraunhofer Institute IAO und IPA Anfang Juli ein Symposium zum Thema Machine Learning für Quantencomputing in Heilbronn.²⁵

Die intensive Zusammenarbeit der beiden Fraunhofer Institute in diesem Projekt hat dazu beigetragen, dass eine gemeinsame Struktur zur Bündelung der Quantencomputingkompetenzen der beiden Gruppen gegründet wurde. Das Fraunhofer-Lab für anwendungsorientiertes Quantencomputing Stuttgart-Heilbronn flQship vereint mehrere Themenschwerpunkte. Quantum Machine Learning ist innerhalb des flQship eine eigenständige Forschungsaktivität geworden.²⁶

Zu Lfd. Nr. 3 wird der Wissenstransfer durch Forschungs- und Innovationsberatung von Industriepartnern aus den Bereichen Produktion und Automotive

1. intern durch regelmäßige Workshops und gemeinsame Use Case Bearbeitung, sowie
 2. extern durch das assoziierte industrienähe Schulungsprogramm (z.B. FhG Academy²⁷)
- gesichert.

Weiterhin hat das Projekt unmittelbar zu einer Folgebeauftragung durch die DLR Quantencomputinginitiative geführt (siehe Abschnitt 3d).

Der durch das Projekt erfolgte Kompetenzaufbau hat weiterhin dazu geführt, dass die beiden Institute IPA und IAO zusammen mit den Fraunhofer Instituten IAF und ITWM ein Test- und Beratungszentrum Quantencomputing²⁸ mit initialem Schwerpunkt QML konzipiert haben. Bei erfolgreicher Bewilligung wird das Projekt vom BMBF unterstützt. Der Vorschlag wurde bereits zu einer Vollbeantragung eingeladen.

²⁴ <https://www.fraunhofer.de/de/presse/presseinformationen/2024/dezember-2024/dieter-schwarz-stiftung-und-fraunhofer-bauen-forschungsstandort-heilbronn-aus.html>

²⁵ <https://flqship.eu/en/ml4qt/>

²⁶ <https://flqship.eu>

²⁷ <https://www.academy.fraunhofer.de/de/weiterbildung/quanten/quantum-machine-learning.html>

²⁸ <https://www.bmbf.de/SharedDocs/Bekanntmachungen/DE/2024/09/2024-09-16-Bekanntmachung-Quanten-computing.html>

Zu Lfd. Nr. 4 fließen die in Lfd. Nr. 1 aufgelisteten Publikationen in die Dissertationen von Daniel Pranjic, Horst Stühler, Frederic Rapp und Daniel Mayer ein. Weiterhin wurden eine Reihe von Masterarbeiten im Rahmen des Projekts durchgeführt [7,10,14].

Zu Lfd. Nr. 5 verspricht sich die USU Software AG durch die Integration von AutoML-Ansätzen in ihre KI Anwendung (KI Ticket Routing) eine vereinfachte Einführung der KI bei Kunden und somit eine verbesserte Skalierbarkeit. Aktuell muss die Konfiguration/Training von KI Spezialisten aus der R&D Abteilung durchgeführt werden. Hierdurch verspricht sich die USU einen nachhaltigen wirtschaftlichen Erfolg durch diese Skalierbarkeit und einfachere Handhabung. Dieser Erfolg kommt auch dem Mittelstand zugute, da mehr KI Einführungsprojekte im Bereich IT Service Management parallel durchgeführt werden können.

Zu Lfd. Nr. 6 verspricht sich die IAV GmbH eine Erhöhung des Automatisierungsgrads ihrer internen ML Pipelines.

Zu Lfd. Nr. 7 betrachtet die KEB Automation KG die Projektergebnisse als wertvolle Prototypenbasis für zukünftige Innovationen im Bereich prädiktive Instandhaltung und intelligente industrielle Diagnostik. AutoML kann hier als Basis für zukünftig zu entwickelnde Software dienen.

Zu Lfd. Nr. 8 Da die ursprünglich angestrebte Software-Lösung zur Erkennung von verkippten Teilen noch nicht realisiert wurde aber weiterhin angestrebt ist, wird TRUMPF an dessen Lösung nach Projektende weiterarbeiten. Methodisch wird diese Lösung allerdings mit klassischen ML-Algorithmen angestrebt werden, da sich im Rahmen von diesem Projekt für den betrachteten Use Case der Bildklassifizierung kein wirtschaftlicher Nutzen durch die Verwendung von QML gezeigt hat.

Zu Lfd. Nr. 9 Das im Rahmen des Projekts eingeführte Tool zur Bewertung von Gebrauchtmaschinen hat eine sehr hohe Akzeptanz in der Fachabteilung gefunden. Das Tool wird täglich von den Kollegen zur Bewertung von Einzelmaschinen aber auch zur Bewertung des Gesamtmarktes genutzt. Das Tool wurde bereits um eine Chat Funktion erweitert und ist fester Bestandteil des An- und Verkaufsprozesses. Nach Projektende soll das Tool auch in anderen Abteilungen wie der Neumaschine und bei Rental zum Einsatz kommen und um jeweilige Funktionen erweitert werden.

Während der Durchführung des Vorhabens dem ZE bekannt gewordener Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Nach unserer Kenntnis ist AutoQML derzeit das einzige Framework seiner Art, dass automatisiert vollumfängliche Pipelines für QML erstellt. Die Einstiegshürde für Unternehmen ist dadurch erheblich gesenkt. Durch die open-source Veröffentlichung des Codes ist die Verbreitung in eine breite Öffentlichkeit sichergestellt.

Erfolgte oder geplante Veröffentlichungen des Ergebnisses nach Nr.11.

Im Rahmen des Projekts wurden die folgenden Veröffentlichungen erstellt. Dabei handelt es sich um 8 peer-reviewed Publikationen, eine Submission, ein Konferenzbeitrag, zwei Whitepaper und drei Masterarbeiten.

- [1] D. Pranjic, B. Mummaneni and C. Tutschku, *Quantum Annealing based Feature Selection*, ESANN (2025)
- [2] Berberich, Julian; Fink, Daniel; Pranjic, Daniel; Tutschku, Christian; Holm, Christian: *Training robust and generalizable quantum models*, Physical Review Research 6(4), 043326 (2024)
- [3] D. Klau, H. Krause, D. A. Kreplin, M. Roth, C. Tutschku and M. Zöller, *AutoQML – A Framework for Automated Quantum Machine Learning*, 2023, https://www.digital.iao.fraunhofer.de/content/dam/iao/ikt/de/documents/AutoQML_Framework.pdf
- [4] D. A. Kreplin, M. Willmann, J. Schnabel, F. Rapp, M. Roth, *sQUlearn: A Python Library for Quantum Machine Learning*, IEEE Software, 2025
- [5] F. Rapp, M. Roth, *Quantum Gaussian Process Regression for Bayesian Optimization*, Quantum Machine Intelligence 6, 5, (2024)

- [6] D. Klau, M. Zöller and C. Tutschku, *Bringing Quantum Algorithms to Automated Machine Learning: A Systematic Review of AutoML Frameworks Regarding Extensibility for QML Algorithms*, 2023, <https://doi.org/10.48550/arXiv.2310.04238>
- [7] V. S. Rajput (2023), *Quantum Support Vector Machines of High-Dimensional Data for Image Classification Problems* [Masterarbeit Universität Stuttgart]. Betreuer: D. Pranjic und S. Brandhofer
- [8] D. Basilewitsch, J. Bravo, C. Tutschku, F. Struckmeier (2024). *Quantum Neural Networks in Practice: A Comparative Study with Classical Models from Standard Data Sets to Industrial Images*. 10.48550/arXiv.2411.19276.
- [9] H. Stühler, M.-A. Zöller, D. Klau, A. Beiderwellen-Bedrikow and C. Tutschku, *Benchmarking Automated Machine Learning Methods for Price Forecasting Applications*, 2023, Proceedings of the 12th International Conference on Data Science, Technology and Applications DATA - Volume 1, 30-39, 2023
- [10] T. Fellner (2024): *Master Thesis: Quantum machine learning for time series prediction* [Masterarbeit Universität Stuttgart]. Betreuer: D. Kreplin, C. Holm (<http://dx.doi.org/10.18419/opus-15247>)
- [11] T. Fellner, D. Kreplin, S. Tovey and C. Holm: *Quantum vs. classical: A comprehensive benchmark study for predicting time series with variational quantum machine learning*. arXiv preprint arXiv:2504.12416 (2025) (submitted to NeurIPS 2025)
- [12] M. Roth, D. A. Kreplin, D. Basilewitsch, J. F. Bravo, D. Klau, M. Marinov, D. Pranjic, P. Schichtel, H. Stühler, M. Willmann, M.-A. Zöller. *AutoQML: A Framework for Automated Quantum Machine Learning*, arXiv:2502.21025 [quant-ph] (2025, akzeptiert bei IEEE QSW)
- [13] P. A. Matt, R. Ziegler, D. Brajovic, M. Roth, M. F. Huber; *A Nested Genetic Algorithm for Explaining Classification Data Sets with Decision Rules*, 2022, <https://doi.org/10.48550/arXiv.2209.07575>
- [14] F. Rapp (2022), *Quantum Gaussian Processes for Bayesian Optimization* [Masterarbeit Universität Stuttgart]. Betreuer: M. Roth, A. Mielke und T. Ricken
- [15] H. Stühler, D. Pranjic, C. Tutschku, *Evaluating Quantum Support Vector Regression Methods for Price Forecasting Applications*, ICAART (3), 376-384 (2024)