Gerlich System and Software Engineering

BMBF-FKZ 50 RA 2013-EVPoC-VeriKI-Schlussbericht-GSSE

Ausgabe : 1 **Version :** 0
Date : 17.07.2023

VeriKI

Verifikation von KI-Methoden in dezentralen Strukturen für New-Space Anwendungen

Teilvorhaben

Entwicklung Verifikationskonzept und Proof-of-Concept (EVPoC)

Gefördert durch:

Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages

BMWi / DLR Fördervorhaben 50 RA 2013

Freigabe	
Name :	Rainer Gerlich
Datum :	17.07.2023
Unterschrift :	

Projektleiter

Ralf Gerlich

Dieses Dokument wurde im Rahmen des BMWK / DLR-Verbundvorhabens
“ VeriKI – Verifikation von KI Methoden in dezentralen Strukturen für New Space*Anwendungen* –
Teilvorhaben: Entwicklung Verifikationskonzept und Proof-of-Concept“ (VeriKI- EVPoC) erstellt.

Inhaltsverzeichnis

Dieses Dokument wurde im Rahmen des BMWK / DLR-Verbundvorhabens
“ VeriKI – Verifikation von KI Methoden in dezentralen Strukturen für New Space*Anwendungen* –
Teilvorhaben: Entwicklung Verifikationskonzept und Proof-of-Concept“ (VeriKI- EVPoC) erstellt.

Dieses Dokument enthält den Schlussbericht zu dem Zuwendungsprojekt FKZ 50 RA 2013-EVPoC

„Verifikation von KI-Methoden in dezentralen Strukturen für New-Space Anwendungen

Teilvorhaben

Entwicklung Verifikationskonzept und Proof-of-Concept (EVPoC)“

gemäß NKBF98, Nr. 8.2.

Dieses Dokument wurde im Rahmen des BMWK / DLR-Verbundvorhabens
“ VeriKI – Verifikation von KI Methoden in dezentralen Strukturen für New Space*Anwendungen* –
Teilvorhaben: Entwicklung Verifikationskonzept und Proof-of-Concept“ (VeriKI- EVPoC) erstellt.

I. TEIL KURZFASSUNG

I.1 AUFGABENSTELLUNG

Das Vorhaben betrifft den Einsatz von Software, die nicht nach den anzuwendenden Standards, bspw. nach dem Standard ECSS der „European Cooperation for Space Standardization“ (ECSS), entwickelt wurde, in sicherheitskritischen Systemen. Diese Software soll Nicht-Konformante Software (Englisch: Non-Conformant Software) genannt und für sie das Akronym NCSW verwendet werden.

Im Vorhaben werden sowohl klassische Software, für die Programmiersprachen wie C, C++, Java verwendet werden, als auch Anwendungen der Künstlichen Intelligenz (KI) betrachtet, speziell lernende Systeme, bezeichnet als Machine Learning (ML) oder Deep Learning auf der Basis neuronaler Netze (NN).

Ziel des Vorhabens ist, einen Prozess zu definieren, durch dessen Anwendung verifiziert werden kann, dass die für ein Projekt ausgewählte NCSW in sicherheitskritischen Systemen eingesetzt werden kann, und auf Beispiele de Prozess anzuwenden.

I.2 VORAUSSETZUNGEN

Das Vorhaben wurde initiiert, weil es große Mengen an NCSW gibt, deren Einsatz in sicherheitskritischen Systemen eine effiziente Realisierung ermöglichen würde, deren Nachentwicklung gemäß anzuwendender Standards aber hohe Kosten und erneute und lange Entwicklungszeiten erfordern würde. Getrieben wird dieser Bedarf u.a. durch kommerzielles Interesse an Raumfahrt, deren Ansatz als NewSpace bezeichnet wird, mit dem Ziel Entwicklung zu beschleunigen und Kosten zu senken.

Aktuell ist der Ansatz von ESA und NASA bei klassischer Software, kleine Teile von NCSW nachzuentwicklen, was auch bereits hohe Kosten bei langer Entwicklungszeit verursacht. Deshalb wird nach einem alternativen Ansatz gesucht, der es ermöglicht, ohne Beschränkung des Umfangs und ohne erneute Entwicklung NCSW zu verwenden. Im Rahmen eines solchen Ansatzes muss verifiziert werden, dass die ausgewählte Software verwendet werden kann.

Die Verifikation von KI-Anwendungen ist i.W. gegenwärtig noch Forschungsthema. Durch die Nutzung von KI in sicherheitskritischen Systemen werden Vorteile erwartet wie kürzere Entwicklungszeiten und geringere Komplexität der Entwicklung. KI-basierte Systeme werden jedoch vollkommen anders entwickelt als klassische Softwaresysteme, so dass wesentliche Teile der üblichen Qualitätssicherungsverfahren auf KI keine oder nur beschränkt Anwendung finden können. Daher soll ein Weg gefunden werden, der beim Einsatz von KI / ML einen sicheren Betrieb ermöglicht.

iBOSS ist ein von der deutschen Raumfahrtagentur geförderter Ansatz für einen modular aufgebauten Satelliten aus standardisierten Würfeln, wobei jeder Würfel eine bestimmte Funktionalität für den Betrieb des Satelliten bereitstellt. Jeder Würfel hat auf jeder seiner sechs Seiten dieselben Schnittstellen zur Umgebung (Struktur, Thermal, Power, Daten), so dass bei bei einer 3-dimensionalen Anordnung bereits viele Kommunikationspfade für Ende-zu-Ende-Verbindungen möglich sind. Bei Ausfall eines Knotens oder einer Verbindung ergeben sich entsprechend viele Möglichkeiten für einen alternativen Pfad. Solche Pfade vorab zu definieren, erfordert einen hohen Planungsaufwand. Daher ergibt sich ein Bedarf für ein sich selbst dynamisch konfigurierendes Netzwerk, das initial die Verbindungspfade findet und bei Ausfall einen alternativen Pfad. Ein solcher dynamischer Ansatz bringt – neben dem Implementierungsaufwand – auch zusätzliche Verifikationsaufgaben für einen sicheren Einsatz mit sich.

Im Teilvorhaben von GSSE werden Erfahrung mit Standards und Verifikation von Software genutzt, um einen alternativen Ansatz für einen Prozess zu den von den Standards anzuwendenden zu finden.

Dieses Dokument wurde im Rahmen des BMWK / DLR-Verbundvorhabens
“ VeriKI – Verifikation von KI Methoden in dezentralen Strukturen für New Space*Anwendungen* –
Teilvorhaben: Entwicklung Verifikationskonzept und Proof-of-Concept“ (VeriKI- EVPoC) erstellt.

I.3 PLANUNG UND ABLAUF

Für das Vorhaben wurden 5 technische Arbeitspakete und ein Management-Arbeitspaket mit einer Gesamtlaufzeit von 24 Monaten geplant. Der technische Fokus des Vorhabens lag auf verteilten Echtzeitsystemen (dezentrale Strukturen) und KI-Anwendungen.

Die technischen Arbeitspakete deckten ab:

- Anforderungen an moderne, dezentrale Raumfahrtsysteme,
  GSSE:
  Standards für Raumfahrtsoftware, Definition von Zuverlässigkeitsanforderungen, Recherche zu bekannten Ansätzen zum Einsatz von NCSW
- Anforderungen an echtzeitfähige verteilte Software,
  GSSE:
  Unterstützung der Partner bzgl. Zuverlässigskeitsanforderungen und Anforderungen bzgl. späterer Verifizierbarkeit
- Einsatz von KI-basierter, lernender Software für NewSpace-Anwendungen,
  Definition von 3 Use Cases (KI-basierte Planung und Lageregelung, dynamische Rekonfiguration)
  GSSE:
  Definition eines Ansatzes zur Mitigation bei auftretenden Störungen und dessen Verifizierbarkeit bzgl. der beabsichtigen Demonstratoren
- Software-Tools und Prüfmethoden, und
  GSSE:
  Umsetzung des Verifikationskonzeptes für die geplante Plattform mit geeigneten Werkzeugen
- Verifikation und Demonstratoren.
  GSSE:
  Analyse des praktischen Einsatzes im Rahmen der beiden Demonstratoren

Das Projekt wurde aus den folgenden Gründen von 24 auf 30 Monate verlängert und das Budget aufgestockt:

- Die bei den Partnern FZI und Universität Würzburg aufgetretenen Verzögerungen führten auch bei GSSE zu einem späteren Beginn konkreter Arbeiten.
- Die Definition eines Prozesses für die Verifikation nicht ECSS-konformer Software (Open Source Software und KI-Software) hat sich als aufwändiger dargestellt als ursprünglich angenommen.
- Die Analyse zur Verifikation des RSTP-Protokolls für dynamisch rekonfigurierbare Netzwerke war komplexer als zunächst angenommen.
- Die Anzahl der zu untersuchenden Szenarien bei der Verifikation von KI hat sich gegenüber den ursprünglichen Annahmen erhöht.
- Die ursprünglich nicht geplante, aber auch aus Sicht von GSSE erforderliche Abstimmung mit RFA-PS (Produktsicherung Software) hat zu zusätzlichem Aufwand geführt.

Zunächst wurde versucht, für klassische Software und KI soweit möglich einen gemeinsamen Prozess zu definieren. Aus Gründen der Übersichtlichkeit wurden dann aber zwei separate Prozesse definiert, die in zwei getrennten Dokumenten beschrieben werden. In den Dokumenten werden auch die Vor- und Nachteile bestimmter Vorgehensweisen abgewogen.

Für den Prozess für klassische Software wurde i.W. auf Erfahrungen und Kenntnisse von GSSE bzgl. Verifikation von Software und potenziellen Problemen zurückgegriffen. Für den Prozess für KI-Anwendungen wurden zunächst Literaturrecherchen durchgeführt, die Ergebnisse analysiert

Dieses Dokument wurde im Rahmen des BMWK / DLR-Verbundvorhabens
“ VeriKI – Verifikation von KI Methoden in dezentralen Strukturen für New Space*Anwendungen* –
Teilvorhaben: Entwicklung Verifikationskonzept und Proof-of-Concept“ (VeriKI- EVPoC) erstellt.

und dann basierend auf Erfahrungen von GSSE bzgl. Zuverlässigkeit zu einem Konzept zusammengeführt.

Der Prozess für klassische Software wurde auf die Software für die dynamische Rekonfiguration angewendet. Der Prozess für KI-Anwendungen wurde von den Partnern für den jeweiligen Use Case angewendet und die Vorgehensweise mit GSSE abgestimmt sowie die Erfahrungen zusammen mit GSSE ausgewertet.

Die Prozessdefinitionen wurden im Rahmen von Diskussionen mit der Software-Produktsicherung der Agentur iterativ optimiert.

I.4 WISSENSCHAFTLICHER UND TECHNISCHER STAND

Der Einsatz von NCSW in sicherheitskritischen Systemen wird aufgrund der Kostensituation und Entwicklungsdauer bei Anwendung etablierter Standards wie ECSS, zunehmend begrenzter Budgets und Nachfrage nach kürzeren Lieferzeiten und dem Druck durch NewSpace sowie an Vielfalt der durch vorhandene NCSW abgedeckten Lösungen zunehmend relevant.

Ansätze wie die Nachentwicklung von (geringen) Teilen von klassischer NCSW nach anzuwendenden Standards können nicht den Bedarf an kostengünstigen, schnell verfügbaren Lösungen abdecken. Daher werden Prozesse benötigt, durch die der Nachweis über zusätzliche und im Umfang begrenzte Verifikationsmaßnahmen erbracht werden kann, dass NCSW in sicherheitskritischen Anwendungen eingesetzt werden können oder mit denen Ausschlussgründe und Wege zu deren Beseitigung identifiziert werden können. Soweit bekannt gab es vor dem Start von VeriKi noch keinen Prozess, der diese Anforderungen unterstützt. Zu Details s. Kap. II.1.1

Bei KI-Anwendungen, speziell beim ML, ist die Verifikation noch Forschungsthema. Ansätze, die die speziellen Anforderungen der Raumfahrt wie zuverlässige Fehlererkennung unterstützen, sind laut der durchgeführten Recherchen nicht bekannt. Zu Details s. Kap.II.1.2.

I.4.1 Angaben zu genutzten Verfahren

I.4.1.1 Klassische NCSW und Verifikation der dynamischen Rekonfiguration

Es wurden Checklisten, Instrumentierung des NCSW-Codes zur Beobachtung der Eigenschaften und Einbettung in Wrapper-Funktionen zur Identifikation und Maskierung von Programmfehlern angewendet. Für die Instrumentierung und Erzeugung der Wrapper-Funktionen wurde das Werkzeug DCRTT von GSSE eingesetzt.

I.4.1.2 KI/ML-Anwendungen

GSSE hat für diese Anwendungen nur beratende Funktion und hat selbst keine KI-Werkzeuge eingesetzt.

I.4.2 Angaben zu genutzter Literatur

Im Folgenden werden Literaturreferenzen aufgeführt, die in den Dokumenten enthalten sind für

- den Prozess für Klassische Software (VeriKI-BSSE-TN09)
- den Prozess für KI-Anwendungen (VeriKI-GSSE-TN10)
- den Testbericht zur dynamischen Rekonfiguration (VeriKI-GSSE-TR01)

Für jedes Dokument werden die Referenzen separat aufgezählt.

I.4.2.1 Prozess für Klassische Software

[1] Cooperation for Space Standardization, ECSS-Q-HB-80-01-A: Reuse of existing software, 2011.

[2] European Cooperation for Space Standardization, ECSS-Q-ST-80-01: Software Product Assurance, 2017

[3] European Cooperation for Space Standardization, ECSS-E-ST-40C: Space engineering - Software, 2009.

[4] DO-178C, Software Considerations in Airborne Systems and Equipment Certification. 2011, RTCA Inc.

1. Euopean Cooperation for Space Standardization, Glossary of terms. 2012. [Online]. Verfügbar unter: https://ecss.nl/standard/ecss-s-st-00-01c-glossary-of-terms-1-october-2012/
2. ECSS Q-ST 40C Safety
3. ECSS-Q-ST-40C Rev.1 – Safety (15 February 2017)
4. ECSS-Q-ST-30C Rev.1 – Dependability (15 February 2017)
5. D. Pangercic, *How Apex.AI Certified ROS 2 According to ISO 26262 ASIL-D,* 2021.
6. https://www.apex.ai/event-details/how-ros-2-was-safety-certified-for-automotive (zuletzt zugegriffen am 07.07.2023)
7. H. Silva, J. Souza, D. Freitas, S. Faustino, A. Constantino und M. Coutinho, „RTEMS Improvement-Space Qualification of RTEMS Executive," in *12t Simpósio de Informática-INFORUM*, Universität Lissabon, 2009.
8. https://www.esa.int/Enabling_Support/Space_Engineering_Technology/Software_Systems_Engineering/RTEMS_EDISOFT (zuletzt zugegriffen am 07.07.2023)
9. „TerraSAR-X, German radar satellite launch successful", EARSC, 15. Juni 2007. https://earsc.org/2007/06/15/terrasar-x-german-radar-satellite-launch-successful/ (zuletzt zugegriffen 25. Januar 2022).
10. „Europa will jetzt auf Kometen landen". https://www.esa.int/Space_in_Member_States/Germany/Europa_will_jetzt_auf_Kometen_landen (zuletzt zugegriffen 25. Januar 2022).
11. M. Verhoef und J. F. Salgado, *RTEMS-SMP Qualification - Status,* 2021.
12. Qualification of RTEMS Symmetric Multiprocessing (SMP)
13. https://essr.esa.int/project/rtems-smp-qdp-qualification-data-package (zuletzt zugegriffen am 07.07.2023)
14. B. Boehm, Software engineering economics. Prentice-Hall, 1981.
15. R. Gerlich, R. Gerlich, A. Fischer, und M. Pinto, „Evaluierung von Software-Verifikations-Werkzeugen: Final Report", 2016.
16. R. Gerlich und R. Gerlich, „Fortführende Evaluierung von Software-Verifikations-Werkzeugen: Final Report", 2017.
17. R. Gerlich, M. Hernek, und R. Gerlich, „Sources of False Positives in Dynamic and Static Analysis and Counter Measures", 2021.
18. H. G. Rice, „Classes of recursively enumerable sets and their decision problems", Transactions of the American Mathematical Society, Bd. 74, S. 358–366, 1953.

[5] FAST Benchmark Report, FASTII-TR1, confidential, 2019, ESA Contract No. 4000116014, ESA Final Presentation, December 4th, 2020, R.Gerlich,M.Hernek,R.Gerlich: "Automatic Stimulation on System Level

[6] H. Kress-Gazit u. a., „Formalizing and guaranteeing human-robot interaction", Commun. ACM, Bd. 64, S. 78–84, 2021.

[7] SAE AS6802: Time-Triggered Ethernet. SAE International, 2016.

Dieses Dokument wurde im Rahmen des BMWK / DLR-Verbundvorhabens
" VeriKI – Verifikation von KI Methoden in dezentralen Strukturen für New Space*Anwendungen* –
Teilvorhaben: Entwicklung Verifikationskonzept und Proof-of-Concept" (VeriKI- EVPoC) erstellt.

[8] IEEE 802.1Qbv-2015 - Enhancements for Scheduled Traffic. 2015

[9] https://de.wikipedia.org/wiki/Ariane_V88 (zuletzt zugegriffen am 07.07.2023)

I.4.2.2 Prozess für KI-Anwendungen

[1] H. Lutz und W. Wendt, Taschenbuch der Regelungstechnik: mit MATLAB und Simulink, 8. Aufl. Frankfurt am Main: Harri Deutsch, 2010.

[2] D. E. Rumelhart, G. E. Hinton, und R. J. Williams, „Learning representations by back-propagating errors“, Nature, Bd. 323, Nr. 6088, S. 533–536, Okt. 1986, doi: 10.1038/323533a0.

[3] G. Katz, C. Barrett, D. L. Dill, K. Julian, und M. J. Kochenderfer, „Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks“, in Computer Aided Verification, Cham, 2017, S. 97–117. doi: 10.1007/978-3-319-63387-9_5.

[4] J. Lunze, Regelungstechnik. 2: Mehrgrößensysteme, digitale Regelung; mit 55 Beispielen, 101 Übungsaufgaben sowie einer Einf. in das Programmsystem MATLAB, 6., neu Bearb. Aufl. Berlin Heidelberg: Springer, 2010.

[5] L. Fahrmeir, Hrsg., Statistik: der Weg zur Datenanalyse, 7. Aufl., korr. Nachdr. Berlin Heidelberg: Springer, 2011.

[6] J. E. van Engelen und H. H. Hoos, „A survey on semi-supervised learning“, Mach Learn, Bd. 109, Nr. 2, S. 373–440, Feb. 2020, doi: 10.1007/s10994-019-05855-6.

[7] L. P. Kaelbling, M. L. Littman, und A. W. Moore, „Reinforcement Learning: A Survey“, Journal of Artificial Intelligence Research, Bd. 4, S. 237–285, Mai 1996, doi: 10.1613/jair.301.

[8] X. Tu, K. Lai, und S. Yanushkevich, „Transfer Learning on Convolutional Neural Networks for Dog Identification“, in 2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS), Nov. 2018, S. 357–360. doi: 10.1109/ICSESS.2018.8663718.

[9] G. Branwen, „The Neural Net Tank Urban Legend“, Sep. 2011, zuletzt zugegriffen: 27. Januar 2022. [Online]. Verfügbar unter: https://www.gwern.net/Tanks

[10] C. Runge, „Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten“, Zeitschrift für Mathematik und Physik, Bd. 46, S. 224–243, 1901.

[11] J. Hochreiter, „Untersuchungen zu dynamischen neuronalen Netzen“, Diplomarbeit, Technische Universität München, 1991.

[12] A. L. Maas, A. Y. Hannun, und A. Y. Ng, „Rectifier nonlinearities improve neural network acoustic models“, 2013.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, und R. Salakhutdinov, „Dropout: a simple way to prevent neural networks from overfitting“, J. Mach. Learn. Res., Bd. 15, Nr. 1, S. 1929–1958, Jan. 2014.

[14] J. Kim, R. Feldt, und S. Yoo, „Guiding deep learning system testing using surprise adequacy“, in Proceedings of the 41st International Conference on Software Engineering, Montreal, Quebec, Canada, Mai 2019, S. 1039–1049. doi: 10.1109/ICSE.2019.00108.

[15] L. Ma u. a., „DeepMutation: Mutation Testing of Deep Learning Systems“, 2018.

[16] W. Shen, J. Wan, und C. Zhenyu, „MuNN: Mutation Analysis of Neural Networks“, 2018.

[17] G. Jahangirova und P. Tonella, „An Empirical Evaluation of Mutation Operators for Deep Learning Systems“, 2020.

[18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, und A. Torralba, „Learning Deep Features for Discriminative Localization“, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Juni 2016, S. 2921–2929. doi: 10.1109/CVPR.2016.319.

[19] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, und D. Batra, „Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization“, in 2017 IEEE International Conference on Computer Vision (ICCV), Okt. 2017, S. 618–626. doi: 10.1109/ICCV.2017.74.

[20] R. Ashmore, R. Calinescu, und C. Paterson, „Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges“, ACM Computing Surveys, Bd. 54, 2021, doi: 10.1145/3453444.

[21] L. Gauerhof, R. Hawkins, C. Picardi, C. Paterson, Y. Hagiwara, und I. Habli, „Assuring the Safety of Machine Learning for Pedestrian Detection at Crossings“, Lissabon, Portugal, 2020. doi: 10.1007/978-3-030-54549-9_13.

[22] M. Zhang, Y. Zhang, L. Zhang, C. Liu, und S. Khurshid, „DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems“, in Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, New York, NY, USA: Association for Computing Machinery, 2018, S. 132–142. zuletzt zugegriffen: 14. Februar 2022. [Online]. Verfügbar unter: https://doi.org/10.1145/3238147.3238187

[23] „MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges“. http://yann.lecun.com/exdb/mnist/ (zuletzt zugegriffen 14. Februar 2022).

[24] ISO/IEC/IEEE 15026-1:2019 - Systems and software engineering — Systems and software assurance — Part 1: Concepts and vocabulary. zuletzt zugegriffen: 14. Februar 2022. [Online]. Verfügbar unter: https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/07/35/73567.html

[25] L. Beining, „KI in der Industrie absichern & prüfen“, S. 21, 2021.

[26] M. Kläs, R. Adler, L. Jöckel, J. Groß, und J. Reich, „Using Complementary Risk Acceptance Criteria to Structure Assurance Cases for Safety-Critical AI Components“, Montreal, Canada, 2021.

[27] M. Kläs, R. Adler, I. Sorokos, L. Jöckel, und J. Reich, „Handling Uncertainties of Data-Driven Models in Compliance with Safety Constraints for Autonomous Behaviour“, München, 2021. doi: 10.1109/EDCC53658.2021.00021.

[28] H.-D. Tran u. a., „NNV: The Neural Network Verification Tool for Deep Neural Networks and Learning-Enabled Cyber-Physical Systems“, Los Angeles, USA, 2020.

[29] S. Bak, H.-D. Tran, K. Hobbs, und T. T. Johnson, „Improved Geometric Path Enumeration for Verifying ReLU Neural Networks“, in Computer Aided Verification, Cham, 2020, S. 66–96. doi: 10.1007/978-3-030-53288-8_4.

[30] Y. Y. Elboher, J. Gottschlich, und G. Katz, „An Abstraction-Based Framework for Neural Network Verification“, in Computer Aided Verification, Cham, 2020, S. 43–65. doi: 10.1007/978-3-030-53288-8_3.

[31] D. J. Fremont, J. Chiu, D. D. Margineantu, D. Osipychev, und S. A. Seshia, „Formal Analysis and Redesign of a Neural Network-Based Aircraft Taxiing System with VerifAI“, in Computer Aided Verification, Cham, 2020, S. 122–134. doi: 10.1007/978-3-030-53288-8_6.

[32] G. Katz, C. Barrett, D. Dill, K. Julian, und M. Kochenderfer, „Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks“, arXiv:1702.01135 [cs], Mai 2017, zuletzt zugegriffen: 25. Januar 2022. [Online]. Verfügbar unter: http://arxiv.org/abs/1702.01135

[33] Zertifizierte KI“, Zertifizierte KI. https://www.zertifizierte-ki.de/ (zuletzt zugegriffen 14. Februar 2022).

[34] „Sicherheits-Prüfungen von KI-Systemen | TÜViT“. https://www.tuvit.de/de/innovationen/ki/ (zuletzt zugegriffen 14. Februar 2022).

[35] H. Kress-Gazit u. a., „Formalizing and guaranteeing human-robot interaction“, Commun. ACM, Bd. 64, S. 78–84, 2021.

[36] „In-flight upset, 154 km west of Learmonth, WA, 7 October 2008, VH-QPA, Airbus A330-303“, Australian Transport Safety Bureau, ATSB TRANSPORT SAFETY REPORT AO-2008-070, Juni 2009.

[37] A. Stocco, M. Weiss, M. Calzana, und P. Tonella, „Misbehaviour Prediction for Autonomous Driving Systems“, Seoul, Südkorea, 2020.

[38] K. Pei, Y. Cao, J. Yang, and S. Jana, ‘DeepXplore: automated whitebox testing of deep learning systems’, *Commun. ACM*, vol. 62, no. 11, pp. 137–145, Oktober 2019, doi: 10.1145/3361566.

[39] Y. Sun, X. Huang, D. Kroening, J. Sharp, M. Hill, and R. Ashmore, ‘DeepConcolic: Testing and Debugging Deep Neural Networks’, in *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, May 2019, pp. 111–114. doi: 10.1109/ICSE-Companion.2019.00051.

[40] L. Ma *et al.*, ‘DeepGauge: multi-granularity testing criteria for deep learning systems’, Montpellier, France, 2018.

[41] G. Jahangirova and P. Tonella, ‘An Empirical Evaluation of Mutation Operators for Deep Learning Systems’, 2020.

[42] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, ‘Stable-Baselines3: Reliable Reinforcement Learning Implementations’, *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[43] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, ‘Proximal Policy Optimization Algorithms’. arXiv, Aug. 28, 2017. doi: 10.48550/arXiv.1707.06347.

[44] G. Brockman *et al.*, ‘OpenAI Gym’. arXiv, Jun. 05, 2016. doi: 10.48550/arXiv.1606.01540.

[45] V. Mnih *et al.*, ‘Playing Atari with Deep Reinforcement Learning’. arXiv, Dec. 19, 2013. doi: 10.48550/arXiv.1312.5602.

[46] V. Mnih *et al.*, ‘Human-level control through deep reinforcement learning’, *Nature*, vol. 518, no. 7540, Art. no. 7540, Feb. 2015, doi: 10.1038/nature14236.

[47] D. Silver *et al.*, ‘Mastering the game of Go with deep neural networks and tree search’, *Nature*, vol. 529, no. 7587, Art. no. 7587, Jan. 2016, doi: 10.1038/nature16961.

[48] D. Silver *et al.*, ‘Mastering the game of Go without human knowledge’, *Nature*, vol. 550, no. 7676, Art. no. 7676, Oct. 2017, doi: 10.1038/nature24270.

[49] B. Grzesik *et al.*, ‘InnoCube—A Wireless Satellite Platform to Demonstrate Innovative Technologies’, *Aerospace*, vol. 8, no. 5, Art. no. 5, May 2021, doi: 10.3390/aerospace8050127.

[50] S. Montenegro *et al.*, ‘InnoCubE Der erste Drahtloser Satellit’, p. 6 pages, 2022, doi: 10.25967/570007.

[51] M. Goeller, J. Oberlaender, K. Uhl, A. Roennau, and R. Dillmann, ‘Modular robots for on-orbit satellite servicing’, in *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Guangzhou, China, Dec. 2012, pp. 2018–2023. doi: 10.1109/ROBIO.2012.6491265.

[52] M. Kortmann, A. Dafnis, T. A. Schervan, H. G. Schmidt, S. Rühl, and J. Weise, 'Building Block-Based "iBOSS" Approach: Fully Modular Systems with Standard Interface to Enhance Future Satellites', presented at the 66rd International Astronautical Congress, Jerusalem, Israel, 2015.

[53] V. Mnih *et al.*, 'Human-level control through deep reinforcement learning', *Nature*, vol. 518, no. 7540, Art. no. 7540, Feb. 2015, doi: 10.1038/nature14236.

I.4.2.3 Verifikation der NCSW für dynamische Rekonfiguration

[1] European Cooperation for Space Standardization, *ECSS-E-ST-40C: Space engineering - Software,* 2009.

[2] European Cooperation for Space Standardization, *ECSS-Q-HB-80-01-A: Reuse of existing software,* 2011.

[3] DO-178C, Software Considerations in Airborne Systems and Equipment Certification. 2011, RTCA Inc.

[4] FZI, „iBOSS – Intelligentes Baukastensystem für das On-Orbit-Satelliten-Servicing und -Assembly", FZI Forschungzentrum Informatik. https://www.fzi.de/forschung/projekt-details/iboss/ (zuletzt zugegriffen 10. Mai 2021).

[5] CycloneSTP | Embedded STP & RSTP for STM32, PIC32, KSZ8863, KSZ8873, LAN9303". https://oryx-embedded.com/products/CycloneSTP.html (zuletzt zugegriffen 24. Januar 2022).

[6] IEEE Standards Association, IEEE 802.1w-2001 - IEEE Standard for Information Technology - Rapid Reconfiguration, IEEE Standards Association, 2001

[7] IEEE 802.1Q-2014 - IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks. 2014.

[8] Dynamic Random Test Tool DCRTT, https://gsse.biz/products/DCRTT/ (zuletzt zugegriffen am 07.Juli 2023)

[9] R.Gerlich,R.Gerlich, veriKI-BSSE-TN08, User Manual for Integration Testing and Use of Wrapper Functions, V1.1, 25.11.2022

[10] IEEE 802.1D. STP

[11] RODOS, Realtime Onboard Dependable Operating System, https://www.informatik.uni-wuerzburg.de/aerospaceinfo/forschung-und-entwicklung-prof-dr-marco-schmidt-kopie-1/wissenschaft-forschung-deprecated/rodos/ (zuletzt zugegriffen am 07. Juli20233)
https://gitlab.com/rodos/rodos (zuletzt zugegriffen 24. Januar 2022)

[12] R.Gerlich, R.Gerlich, veriKI-BSSE-TN05, Verifikation des Routers, Version 1.1, 29.03.2023)

I.5 ZUSAMMENARBEIT

Das Vorhaben war ein Verbundprojekt von FZI, Universität Würzburg und GSSE. Die Zusammenarbeit von GSSE mit FZI und Universität Würzburg erfolgte

- zur Unterstützung und Beratung bzgl. Zuverlässigkeitsanforderungen der Raumfahrt,
- Verifizierbarkeit der Software der Use Cases,
- Anwendung des KI-Prozesses für den jeweiligen Use Case, und
- Diskussion der Ergebnisse nach Anwendung und Empfehlungen für Verbesserungen.

II. DETAILLIERTE DARSTELLUNG

II.1 VERWENDUNG DER ZUWENDUNG

Die Zuwendung wurde verwendet, um

- Prozesse zu definieren, die eine Verwendung von NCSW in sicherheitskritischen Systemen durch Verifikation ermöglichen, und
- im Rahmen von praktischen Anwendungen die Machbarkeit und Vorteile zu demonstrieren.

II.1.1 Der Prozess für klassische Software

Der Prozess besteht aus folgenden wesentlichen Schritten:

- Abarbeitung von Checklisten,
- Einbettung der NCSW zur Mitigation bei potenziellen Problemen, und
- Beobachtung der Eigenschaften der NCSW im praktischen Einsatz und der Auswertung der Ergebnisse.

Durch die Checklisten soll eine Vorauswahl von Maßnahmen getroffen werden anhand von verschiedenen Kriterien. Sofern durch die Checklisten kein Ausschlusskriterium identifiziert wurde, werden im nächsten Schritt Mitigationsmaßnahmen und Maßnahmen zur Beobachtung der Eigenschaften im Betrieb definiert. Im dritten und letzten Schritt werden dann durch Ausführung der NCSW deren Eigenschaften beobachtet und bewertet. Fällt die Bewertung positiv aus, dann kann die NCSW in sicherheitskritischer Umgebung eingesetzt werden.

Parallel zu den genannten Schritten wird die Dokumentation der Verantwortlichkeit bzgl. getroffener Entscheidungen gefordert. Der Prozess kann an jeder Stelle durch Entscheidungsträger abgebrochen werden, die eine Risiko-Nutzen-Abwägung treffen müssen, und für das vorhandene Restrisiko die Verantwortung übernehmen müssen.

Die Endversion des Prozesses ist das Ergebnis von GSSE-Konzepten und iterativer Optimierung durch Diskussionsbeiträge von der Softwareproduktsicherung der Agentur.

II.1.2 Der Prozess für KI-Anwendungen

Für den Prozess für KI-Anwendungen wurden umfangreiche Literatur-Recherchen durchgeführt. Verschiedene Ansätze wurden identifiziert und bewertet. Der abgeleitete Prozess berücksichtigt die unterschiedlichen Randbedingungen und Anforderungen zwischen Raumfahrt und anderen Bereichen.

Sicherheitskritische Raumfahrtanwendungen, bei denen Gefahr für Verletzung oder Leben oder Ausfall der Mission droht, müssen zu einer bestimmten Zeit bzw. in einem bestimmten Zeitfenster korrekt arbeiten. Eine Wiederholung ist in den meisten Fällen nicht möglich. Aus diesen Anforderungen der Raumfahrt kann abgeleitet werden, dass

- Fehler erkannt werden müssen, und
- entweder der Fehler vollständig behoben oder der Übergang in einen sicheren Zustand möglich sein muss.

Die gemeinsame Mindestanforderung ist daher die Möglichkeit der Fehlererkennung.

Es gibt KI-Anwendungen, bei denen nicht spezifiziert werden kann, wann die KI sich fehlerhaft verhält. Dies ist etwa bei der Objekterkennung für das automatisierte Fahren der Fall. Daher werden Fehler toleriert und Maßnahmen ergriffen, die die Fehlerrate reduzieren sollen. Fehlerhaftes Verhalten ist aber möglich und kann nicht grundsätzlich verhindert werden. Die Ermittlung der konkreten Fehlerhäufigkeiten ist mit großem Aufwand verbunden, der u.a. reale

Dieses Dokument wurde im Rahmen des BMWK / DLR-Verbundvorhabens
“ VeriKI – Verifikation von KI Methoden in dezentralen Strukturen für New Space*Anwendungen* –
Teilvorhaben: Entwicklung Verifikationskonzept und Proof-of-Concept“ (VeriKI- EVPoC) erstellt.

bzw. sehr realitätsnahe Tests erfordert. Insofern scheiden diese Ansätze für Raumfahrtanwendungen aus.

Formale Verifikation von KI ist ein aktives Forschungsthema. Nur einfache Beispiele mit (sehr) wenig Parametern sind in der Literatur zu finden. Daher scheidet auch dieser Ansatz aus.

Majority-Voting zur Fehlererkennung scheidet aus, da es im Gegensatz zum Voting bei Hardware nicht naheliegt, dass die Majorität die korrekte Lösung repräsentiert. Darüber hinaus setzt Voting mehrere voneinander unabhängige KI-Systeme mit derselben Aufgabe und die Kenntnis eines Vergleichsmaßstabs der von den Systemen vorgeschlagenen Aktionen voraus. Dabei werden die ohnehin schon kostbaren Trainingsdaten zusätzlich auf mehrere Systeme aufgeteilt und es wird forciert, dass alle dieser KI-Systeme dieselben Lösungen für das Problem hervorbringen. Dies ist – z.B. bei Regelungsanwendungen – nicht immer der Fall. Somit scheidet auch dieser Ansatz aus.

Daher verbleibt als einziger Ansatz die Überwachung und Fehlererkennung, um Mitigationsmaßnahmen einleiten zu können. Voraussetzung dafür ist aber, dass ein Fehler erkannt werden kann. Für die beiden Use Cases im Vorhaben ist das möglich.

Kann bei einer Raumfahrtanwendung ein Fehler der KI nicht erkannt werden, dann sollte der Einsatz von KI in diesem Fall ausgeschlossen werden bzw. ist dann mit sehr hohem Aufwand und/oder Risiken verbunden.

Nachvollziehbar ist, dass für die Mitigation nicht erneut KI eingesetzt werden kann. Daher bleibt als einzige Möglichkeit der Einsatz von klassischer Software zur Mitigation. Wenn KI eingesetzt werden soll, dann ist die Erwartung, dass sie wesentliche Vorteile gegenüber einer Lösung mit klassischer Software bringt. Daraus folgt, dass mit der zur Mitigation eingesetzten Software nicht dieselbe Effizienz erreicht werden kann wie durch KI. Ist das der Fall, ist der Einsatz von KI unnötig. Umgekehrt schließt die Notwendigkeit von KI zur Problemlösung aus, dass eine Lösung mit klassischer Software zur Mitigation zur Verfügung steht.

Während der Arbeiten wurde bekannt, dass sich eine ESA-Arbeitsgruppe gebildet hat, die am Handbuch „ECSS-E-HB-40-02A - Machine learning qualification for space applications“ arbeitet. Daher wurde die zum damaligen Zeitpunkt verfügbare Information zum KI-Prozess und den bewerteten Recherchen übermittelt.

Die Ergebnisse wurden außerdem auf der Konferenz „DAta Systems In Aerospace 2023“ im Juni 2023 präsentiert. Ein dort anwesendes Mitglied der Arbeitsgruppe zum ECSS-Handbuch gab im persönlichen Gespräch an, dass die Empfehlungen aus dem VeriKI-Vorhaben im Wesentlichen mit den im seit Frühjahr 2023 im Review befindlichen Handbook übereinstimmen.

Die Endversion des Prozesses ist das Ergebnis von GSSE-Konzepten und iterativer Optimierung durch Diskussionsbeiträge von der Softwareproduktsicherung der Agentur, durch die auch der Kontakt zur ESA-Arbeitsgruppe hergestellt wurde.

II.1.3 Dynamische Rekonfiguration

Eine öffentlich verfügbare Implementierung CycloneSTP des Rapid Tree Spanning Prokolls (RSTP) wurde zusammen mit ergänzender Software der Universität Würzburg auf einer Plattform mit einem Raspberry Pi 4 und Linux mit einer Basisnetzwerkkonfiguration hinsichtlich Codeabdeckung, Ausführungszeiten der Funktionen, Verhalten nach Ausfällen im Netzwerk und Ende-zu-Ende-Ausführungszeiten analysiert. Eine korrekte Rekonfiguration konnte verifiziert werden. Randbedingungen für den Einsatz wurden ebenfalls untersucht und beschrieben. Eine Korrektheitsaussage für eine konkrete Anwendung kann aus diesen Experimenten jedoch nicht unmittelbar abgeleitet werden. Die Tests müssten dazu mit der konkreten Anwendung und der dort verwendeten Netzwerkkonfiguration wiederholt werden.

Dieses Dokument wurde im Rahmen des BMWK / DLR-Verbundvorhabens
“ VeriKI – Verifikation von KI Methoden in dezentralen Strukturen für New Space*Anwendungen* –
Teilvorhaben: Entwicklung Verifikationskonzept und Proof-of-Concept“ (VeriKI- EVPoC) erstellt.

II.1.4 Anwendung der Prozesse

II.1.4.1 Klassische Software

Der Prozess wurde auf die Software für die dynamische Rekonfiguration eines Netzwerkes angewendet. Der Fokus lag dabei auf der Einbettung der Funktionen zur Mitigation und Identifikation von Fehlern während des Betriebs und der Beobachtung der Eigenschaften.

Im Rahmen des Testszenarios traten keine Fehler auf. Durch die Beobachtung der Eigenschaften wurden jedoch Einschränkungen bzgl. Einsatzmöglichkeiten identifiziert. Wesentlich ist, dass diese Einschränkungen nicht durch – ggf. erweiterte – Checklisten gefunden werden können, sondern nur durch eine Analyse der zur Laufzeit beobachteten Eigenschaften.

Daher wird die Analyse der konkret nachweisbaren Eigenschaften als unverzichtbar eingestuft, während die Checklisten nur als mögliches Auswahlkriterium eingestuft werden können, durch die aber ggf. schon in einem frühen Stadium eine Anwendung der NCSW ausgeschlossen werden kann. Ein positives Endergebnis impliziert nicht, dass die NCSW auch für die Anwendung geeignet ist.

II.1.4.2 KI-Anwendungen

Für beide Anwendungen wurde der Prozess angewendet, wobei der Fokus auf Fehlererkennung und -behebung lag. Für den Use Case „Planung“ wurde festgestellt, dass die Fehlererkennung mit einfachen Mitteln über klassische Software möglich ist, aber kein Backup über klassische Software aus Gründen der Komplexität und benötigten Zeit zur Verfügung gestellt werden kann. In diesem Fall ist nur der Übergang in einen sicheren Zustand möglich.

Beim Use Case „Lageregelung“ kann auch durch Anwendung geeigneter Kriterien auf die Ausgangsdaten der KI-Regelung ein Fehler erkannt werden, und es ist ein Weiterbetrieb mit einem klassischen Regler mit geringerer Performance möglich.

Dieses Dokument wurde im Rahmen des BMWK / DLR-Verbundvorhabens
“ VeriKI – Verifikation von KI Methoden in dezentralen Strukturen für New Space*Anwendungen* –
Teilvorhaben: Entwicklung Verifikationskonzept und Proof-of-Concept“ (VeriKI- EVPoC) erstellt.

II.2 ZAHLENMÄßIGER NACHWEIS

Die folgende Tabelle zeigt die Kostenplanung und die tatsächlich entstandenen Kosten für die Gesamtkosten, jeweils für die ursprüngliche Planung von 24 Monaten und die endgültige Planung nach Aufstockung für 30 Monate.

Position	Planung für 24 Monate	Planung für 30 Monate	
	Soll	Soll	Ist
Gesamtkosten	190.905,00 €	242.198,66 €	250.547,88 €
Bundes-/Förderanteil	133,633.50 €	169.539,06 €	169.539,06 €
Eigenanteil	57,271.50 €	72.659,60 €	81.008,82 €
Förderquote	70%	70%	67,7%

Bzgl. Einzelheiten wird auf den Verwendungsnachweis verwiesen.

Corona-bedingt konnten die von 2020 bis Anfang 2022 geplanten Reisen nicht stattfinden. Sie wurden durch Online-Konferenzen ersetzt.

Die Mittel für eine Reise zu einer europäischen Konferenz (Spanien) im Juni 2023, bei der 2 Vorträge zu Ergebnissen von VeriKI gehalten wurden, konnten in Höhe von € 1650,00 nicht abgerufen werden, da der Reisezeitraum außerhalb der Vorhabenslaufzeit lag, die am 28.02.2023 endete. An tatsächlichen Reisekosten innerhalb der Vorhabenslaufzeit fielen an € 709,96 gegenüber geplanten € 6590,00.

An sonstigen Vorhabenskosten waren geplant € 2100,00 – u.a. für die Anmietung von aufgrund der Corona-Beschränkungen nicht benötigten Meetingräumen –, abgerufen wurden €530,00.

Die restlichen Mittel für Reisen und sonstige Kosten wurden auf Antrag in Personalstunden umgewidmet, wodurch die Personalkosten, die über die im Aufstockungsantrag genannte Menge hinuasging, teilweise gedeckt wurden.

II.3 NOTWENDIGKEIT UND ANGEMESSENHEIT

Die Aktivitäten dieses (Teil-)Vorhabens liefern einen Beitrag zur Reduktion von Kosten und Entwicklungszeit für Software bei Wahrung der Anforderungen an Zuverlässigkeit und Sicherheit bei Raumfahrtsystemen. Die Ergebnisse ermöglichen

- im Fall klassischer Software die Erschließung des Potenzials vorhandener Software, die nicht nach Standards entwickelt wurde,
- im Fall von KI / ML deren Nutzung in sicherheitskritischen Systemen.

Sie decken damit einen Bedarf, der hinsichtlich Kommerzialisierung und breiterer, weil kostengünstigerer Anwendung von Raumfahrtsystemen im Rahmen von NewSpace entstanden ist und zukünftig noch wachsen wird.

Bzgl. der dynamischen Netzwerkkonfiguration ist durch das iBOSS-Konzept ein Bedarf entstanden. Die Ergebnisse der Untersuchung im Vorhaben geben potenziellen Anwendern Hinweise, wo die Grenzen des Konzeptes liegen.

II.4 VERWERTBARKEIT

Die Ergebnisse des Vorhabens sind kurz- bis mittelfristig in NewSpace-Projekten sowie in nationalen und internationalen bzw. ESA-Projekten anwendbar.

Die Ergebnisse der Analyse zur dynamischen Netzwerkkonfiguration können bei zukünftigen iBOSS-Missionen oder anderen Missionen, die ein vermaschtes Netzwerk nutzen, verwendet werden.

Der Prozess zu KI soll beim nächsten Satellitenprojekt „InnoCube“ der Universität Würzburg angewendet werden.

Inhalte des KI-Prozesses wurden bereits an die Arbetisgruppe für das ESA-Handbuch zum Machine Learning eingeflossen.

Die Prozesse zur Klassischen Software und zu KI können auch in Bereichen außerhalb der Raumfahrt wie Automotive angewendet werden. Der Prozess zur Klassischen Software ermöglicht, verfügbare Software in sicherheitskritischen Systemen der Kategorie B/C einzusetzen.

II.5 FORTSCHRITTE AUßERHALB DES VORHABENS

Während der Laufzeit des Vorhabens sind keine neuen Ansätze gefunden worden, die äquivalent zu den im Vorhaben erarbeiteten sind. Die Ergebnisse der ECSS-Arbeitsgruppe zum KI-Handbook und die Ergebnisse aus dem Vorhaben stützen sich gegenseitig.

Im Automotive-Bereich laufen Aktivitäten zum Einsatz von KI, wobei der Fokus auf Bildverarbeitung liegt. Die Lösungsansätze haben das Ziel, entweder die Fehlerraten der KI durch Maßnahmen während des Trainings oder durch qualitative Maßnahmen während des Betriebs zu minimieren. Durch die Ansätze kann ein sicherer Systemzustand nicht garantiert werden, wie bei der Raumfahrt erforderlich, so dass vorwiegend noch menschliche Fahrerinnen und Fahrer als Fallback benötigt werden. Die Maßnahmen betreffen die Reduktion eines statistischen Mittelwertes der Fehlerereignisse, während bei Raumfahrtanwendungen im Einzelfall Sicherheit garantiert werden muss, da die Stückzahl nicht hoch ist und/oder keine Möglichkeit der Wiederholung besteht. Ein Assistenz durch Personen wie im Automotive-Bereich ist bei (unbemannten) Raumfahrtsystemen nicht möglich.

Weitere Ansätze wie die Anwendung formaler Methoden befinden sich noch im Forschungsstadium bei einfachen Use Cases und sind daher kurz- bis mittelfristig nicht einsetzbar.

Bzgl. Klassischer Software wird sowohl im Raumfahrtbereich (ESA, NASA) als auch im Automotivebereich weiterhin auf Neuentwicklung von kleinen Teilen der Software gesetzt. Während der Laufzeit des Vorhabens haben sich keine Änderungen ergeben.

Analysen bzgl. der dynamischen Netzwerkkonfiguration und Einsatz unter ähnlichen Bedingungen wie für die Raumfahrt erforderlich, sind nicht bekannt.

II.6 VERÖFFENTLICHUNGEN

[1] Eurospace Symposium DASIA'23 "Data Systems in Aerospace", 06 – 08 June, 2023, Sitges, Spain, R.Gerlich:R.Gerlich,S.Montenegro,F.Puppe,K.Djebko,C.Plasberg,M.Bädorf, "It's the data, stupid! Constructive and analytical quality-assurance for AI-based space systems”

[2] Eurospace Symposium DASIA'23 "Data Systems in Aerospace", 06 – 08 June, 2023, Sitges, Spain, R.Gerlich:R.Gerlich,S.Montenegro,M.Bädorf, “Caught in the net: Reliable adaptive routing in mesh-based on-board networks – are we there yet?”

[3] ESE Kongress 2022, 08.12.2018, Sindelfingen, Germany, R.Gerlich,M.Baedorf,R.Gerlich: "Nutzung von nicht-konformanter Software in kritischen Systemen – Identifikation und Reduktion von Risiken“

[4] Eurospace Symposium DASIA'22 "Data Systems in Aerospace",17 – 19 May, 2022, online-edition, R.Gerlich, F.Flederer,D.Timmermann,R.Gerlich, “Verification of Non-conformant Software (OSS and AI)”