Peter Haber
Thomas J. Lampoltshammer
Helmut Leopold
Manfred Mayr *Eds.*

Data Science – Analytics and Applications

Proceedings of the 4th International Data Science Conference – iDSC2021

EBOOK INSIDE

Springer Vieweg

Data Science – Analytics and Applications

Peter Haber · Thomas J. Lampoltshammer ·
Helmut Leopold · Manfred Mayr
Editors

Data Science – Analytics and Applications

Proceedings of the 4th International Data Science Conference – iDSC2021

Springer Vieweg

Editors

Peter Haber
Salzburg University of Applied Sciences
Salzburg, Austria

Helmut Leopold
AIT Austrian Institute of
Technology GMBH
Vienna, Austria

Thomas J. Lampoltshammer
University for Continuing Education Krems
Krems, Austria

Manfred Mayr
Salzburg University of Applied Sciences
Salzburg, Austria

ISBN 978-3-658-36294-2 ISBN 978-3-658-36295-9 (eBook)
https://doi.org/10.1007/978-3-658-36295-9

Responsible Editor: David Imgrund
This Springer Vieweg imprint is published by the registered company Springer Fachmedien Wiesbaden GmbH part of Springer Nature.
The registered company address is: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

Preface

Based on the overall digitalisation in all spheres of our lives, Data Science and Artificial Intelligence (AI) are nowadays cornerstones for innovation, problem solutions and business transformation. Data, whether structured or unstructured, numerical, textual, or audiovisual, put in context with other data or analysed and processed by smart algorithms, are the basis for intelligent concepts and effective solutions. These solutions are addressing many application areas such as Industry 4.0, Internet of Things (IoT), smart cities, smart energy generation and distribution, and environmental management. Innovation dynamics and business opportunities as effective solutions for the essential societal, environment, or health challenges, are enabled and driven by modern data science approaches.

However, Data Science and Artificial Intelligence are forming a new field that needs attention and focused research. Effective data science is only achieved in a broad and diverse discourse – when data science experts cooperate tightly with application domain experts and scientists exchange views and methods with engineers and business experts. Thus, the **4th International Data Science Conference** (iDSC 2021) brought together researchers, scientists, and business experts to discuss new approaches, methods, and tools made possible by data science.

The cooperation of the Salzburg University of Applied Sciences, the Vorarlberg University of Applied Sciences, the University for Continuing Education Krems (Danube University Krems), and the AIT Austrian Institute of Technology demonstrates the strong Austrian scientific footprint and a deep commitment to a cooperative effort for jointly building an international community from science, research, as well as business, through data science and data analytics.

The iDSC is designed as a conference with a dual approach: By bringing together the latest findings in data science research and innovative implementation examples in business and industry, the conference is aimed at reflecting the current scientific breakthroughs and application expertise, as a means of stimulating shared professional discourse. The six thematic sessions of the conference have been mirroring all the top issues of the data science discipline ranging from challenges in the industrial setting, via Deep and Machine learning methodologies and Natural Language Processing approaches up to future innovation strategies. While the Research Track had a strong emphasis on Safety and Security matters like Anomaly Detection, Integrity Awareness or Ethical fairness of AI e.g., the Industry Track more made deep reflections on the Scaling of Business models, Infrastructure and Software, the Use of Data Science in SMEs, or the development of Data Ecosystems as an Incubator for Data Innovation.

5 keynotes, two from the research arena and three from different industries, had enhanced the conference's total outcome by deliberating on data science from a meta-level: The topics "AI everywhere as a Social good", "The paradigm shift to the circular economy pushed by Smart Data", "The new cultural mindset of Data Mesh", "Open Science with limited closed data-sharing and the development of synthetic data models" and last but not least "Potentials for agile and transparent Data Governance with domain-driven, decentralized Data management" are representing a shining mix of themes – philosophically grounded – making the iDSC an event not to miss in the community.

Therefore - once again - the iDSC, showed that the chosen structure of research and industry tracks provided fascinating insights into current areas of research, as well as presenting impressive use cases to demonstrate the huge potential and significance of modern data science. With our new service, all talks can be now accessed also via video stream from our landing page https://idsc.at.

This was also made possible by the excellent work and outstanding execution of the conference by Michael Mürling, Athina Lykou and the team at AIT, as well as by our technical support team brutkasten.com. We greatly appreciate the generous support of SBA Research, X-NET Services, SPARX Services and as well our media partner Trending Topics. We also want to acknowledge Julian Nöbauer and Maximilian Tschuchnig from the Salzburg University of Applied Sciences for their ongoing backing especially on the creation of the proceedings. Finally, we thank the organizing & program committee as well as the reviewers for helping to make the conference a reoccurring success.

Enjoy the present proceedings of the conference and see you in 2023 at the University for Continuing Education Krems (Danube University Krems).

Peter Haber, Thomas Lampoltshammer, Helmut Leopold, Manfred Mayr
Conference Chairs

Data science & AI depend on smart ecosystems to provide society with innovative solutions

An overview of AI solutions "Made in Austria"

Digitalisation has changed the rules of business and many social mechanisms at an amazing pace. While hugely powerful devices such as smartphones, laptops, and PCs have served to network people on a global scale during the past decade, this transformation process has gained further momentum through the networking of our physical objects to create the Internet of Things (IoT). These developments, in turn, create the potential for new applications, business models, and value chains. However, this has simultaneously made us dependent on technology platforms, to the extent that our economy, our social lives, and our public administration are now all unthinkable without functioning digital infrastructures.

Three challenges must be overcome to ensure that this transformation is beneficial for mankind:

1. Mastery of digital technology platforms has become a fundamental requirement for business and society. Digital technology and infrastructure must be designed for maximum availability and offer the best possible level of security from a wide range of threats. Developments which focus on both minimal resource consumption and data protection in the service of humanity are essential.

2. Establishing extremely high-performance data management and ensuring that data sovereignty remains in the hands of the user is the order of the day. By living and working with a multitude of IoT devices, we continually generate huge data volumes which can be combined and processed using smart algorithms to produce essential information. This allows us to use digital platforms and smart data management to effectively address society's key challenges, including the environment, energy, and mobility. Smart IT services and high-performance computing (HPC) play a key role in determining productivity in our digital future. Contrary to the current cloud megatrend, this will also require new network architectures to balance data transmission and computing power between end devices (IoT) and data centres (cloud).

3. Lastly, effective cooperation must be fostered between data scientists and domain experts. Effective and solution-oriented data science and artificial intelligence (AI) can only function based on new forms of cooperation between the various disciplines. Computer scientists rely on mechanical engineers, electrical engineers, physicists, architects, etc., and vice versa, to successfully develop useful, needs-based, functional data science and AI solutions.

These challenges are a key research focus at the Center for Digital Safety & Security at the AIT Austrian Institute of Technology. Current highly innovative AIT developments "Made in Austria" include:

- **Smart encryption for secure cloud solutions:**

  » Smart data encryption to give data owners dedicated selective and dynamic access, even in distributed cloud systems (e.g., https://secredas-project.eu/ and https://profet.at/).

  » Next-generation data back-up and archiving solutions in public or hybrid cloud storage. Distributed and encrypted data can be stored securely in the cloud, without even the cloud provider being able to access and analyse the stored data (https://www.fragmentix.com/de/).

  - » Virtualized, distributed (blockchain-based) database architectures in the cloud to create new marketplaces. Using a highly secure cloud solution, encrypted supply and demand information in a distributed system can be retrieved automatically and compared anonymously (https://www.flexprod.at/de). This innovation won the German Digital Leader Award 2020 and is marketed through CATCH.direct (https://www.catch.direct/).

- **Cyber security AI solutions and new quality of experience for customers of digital services:**

  - » Modern cyber security solutions must also be able to detect unknown and non-specified threats and attacks on IT systems. New AI-based anomaly detection systems are therefore essential for future security information and event management systems (SIEM) (https://aecid.ait.ac.at/), H. Leopold et al. Cyber Attack Information System – Erfahrungen und Erkenntnisse aus der IKT-Sicherheitsforschung, 2015, Springer Verlag, https://link.springer.com/book/10.1007/978-3-662-44306-4)

  - » Tomorrow's software and system development need new software engineering approaches, particularly for safety-critical systems. This ensures safety & security are factored into the design, allowing effective security certification (https://www.threatget.com/).

  - » Modern network operators need machine learning systems for effective and dynamic network management and to provide the best possible quality of experience to digital end users (https://bigdama.ait.ac.at/).

  - » Effective AI solutions for protection against cyber crime (https://www.fakeshop.at/) and as weapons in the battle against disinformation and fake news (https://www.defalsif.ai).

- **Artificial intelligence and new data economies in the service of mankind as an important contribution to solving important societal challenges:**

  - » Access to the data continually being generated in the digital space (Open Data) as well as new IT system architectures and algorithms are needed to enable new data economies and to support data cooperation and data sharing. Examples include the Austrian Data Intelligence Offensive (DIO) (https://www.dataintelligence.at/) and the European Gaia-X initiative.

  - » The general availability of data about product characteristics, materials and life cycles allows raw materials to be recycled and products to be specifically processed in keeping with the concept of a circular economy.

  - » Shifting AI from the data centre to the edge gives rise to numerous new applications while simultaneously increasing resilience and data security and reducing overall energy consumption.

All these examples show what fascinating innovations are possible when data experts cooperate closely with domain experts, users and authorities and share their expertise in innovation processes and smart, agile design-thinking ecosystems. Ultimately, smart ecosystems are the true drivers of innovation when it comes to developing AI solutions that will benefit humanity.

Helmut Leopold
Head of Center for Digital Safety & Security
AIT - Austrian Institute of Technology

Data boost industry-academia link

The bilateral focus of the iDSC conference is nicely reflected in the progress and results of DataKMU, which is a three-year research and transfer endeavour with participation from the industry as well as from academia including Salzburg University of Applied Sciences. The wide-spread uncertainty in small and medium-sized enterprises as regards utilisation of current data-driven concepts and methodologies to improve their businesses has motivated the DataKMU consortium to provide low-threshold access to a wide variety of state-of-the-art approaches in applied data science. The central goal is to systematically establish a multi-faceted operational industry-academia link in the field of data science, which is presented in the following.

A pivotal step of improvement in any situation is to get a clear picture of one own‘s status quo and to derive from this insight well-suited potential options for enhancement. Therefore, a multi-case study was undertaken to infer criteria for the determination of a so-called Data Science Readiness Level specifically for SMEs. This multidimensional scale, which is in part based on the Data Science Maturity Model from Oracle (described in the article “Strategic Approaches to the Use of Data Science in SMEs” in this Proceedings), allows SMEs to position themselves with little effort and to derive from this proper options for initial quick-wins in advanced data utilisation. This can be considered ramp-up support. SMEs can also locate themselves inside groups such as Practitioners, Strategists, and Pioneers and get related hints and information regarding suggested action points and potential caveats.

Data science strategies and their technical implementation are usually very domain-specific, which is why several showcase prototypes were created as part of the DataKMU project. These tangible results are open to analysis and discussion from various stakeholders, which is specifically interesting for companies that are too small to have their own data science teams on their payroll. The best-practise example implementations such as (i) Transfer Learning for automated data labelling in marketing, (ii) Data Analytics as a Service for virtual sensors as system observer in production, (iii) Development of automated methods for the detection and data extraction of signposts in tourism, and (iv) Automated recording of road conditions in logistics motivate SMEs to adopt similar solutions to stay competitive in their respective markets. Thus, by acting as ‚innovation followers‘ in the beginning, these SMEs can lower their threshold to the level of innovation leaders considerably.

In addition to the above-mentioned example implementations, a rather generic big data pipeline infrastructure was established to support SMEs in their bootstrapping of their own data science applications. This pipeline is a stateful pipeline that ‘remembers’ parametrisations over time in a way that can be re-played and adopted to optimise the various algorithmic building blocks. Thus, variants of solutions and related success can be analysed easily without the usual hardware costs.

The joint operation of several tertiary educational institutions helps to sharpen the respective profiles of data science-related programs in various places in the western part of Austria and adjacent regions of Germany. Replicating similar curricula is not an option to encourage a maximum of potential students to enrol in MINT programs. Thus, the specific strengths of the institutions were identified in course of the DataKMU project and also the main present and future research focus areas representing their characteristics in the individual universities. It is by far better to develop specific fields of application to address a higher total number of young people with study interest in data science. The intended heterogeneity in education also has a broader publicity as a side effect.

One of the pivotal results of the DataKMU endeavour is that the linkage between regional business players and regional universities is a very productive setting for the innovation system. Local businesses domain knowledge combined with methodological competences from research institutions enriched by the creative potential of young students forms an incubator setting that is beneficial to all the various stakeholders likewise. This creates a faster convergence of solutions in a pre-competitive environment.

In addition, a long-term strategic roadmap was developed together with the consortium by an external partner. For this purpose, a catalogue of measures was developed with external expertise to sustainably anchor the DataKMU network regionally, nationally, and internationally. One of the next steps of the consortium is to apply for a follow-up grant in the realm of ecologically-oriented data science for businesses and industries, where several of the above-mentioned strategic benefits will again play an important role, which will also be in line with the European Green Deal initiative. We look forward to presenting new findings at one of the next iDSC conferences.

Thomas J. Heistracher
Department Head Informatics and Software Engineering
Research Director Information Technologies
Salzburg University of Applied Sciences | Fachhochschule Salzburg GmbH

Organization

Organizing Institutions
AIT - Austrian Institute of Technology
Salzburg University of Applied Sciences

Conference Founders and General Chairs
Manfred Mayr
Peter Haber
Thomas Lampoltshammer

Local Conference Chairs
Helmut Leopold
King Ross
Michael Mürling

Organising Committee
Athina Lykou
Gabriela Viale Pereira
Helmut Leopold
Julian Nöbauer
Kathrin Plankensteiner
Manfred Mayr
Maximilian Tschuchnig
Michael Mürling
Peter Haber
Robert Merz
Thomas Lampoltshammer
Valerie Albrecht

Program Committee

Alexandra Posekany
Andreas Unterweger
Andreas Weiler
Arne Bathke
Cornelia Ferner
David Anastasiu
Elena Lloret Pastor
Eric Davis
Felix Strohmeier
Florina Piroi
Frank Danielsen
Gabriela Viale Pereira
Giuseppe Manco
Gregor Eibl
Günther Eibl
Helmut Leopold
Johannes Scholz
Karl Entacher
Karl-Heinz Weidmann
Klaus Frick
Manfred Mayr
Manolis Koubarakis
Mario Drobics
Maximilian Tschuchnig
Michael Gadermayr
Michael Hellwig
Michael Mürling
Peter Haber
Robert Merz
Ross King
Sebastian Hegenbart
Simon Kranzer
Steffen Finck
Thomas Heistracher
Thomas Lampholtshammer
Willibald Krenn

Reviewers

Albin Ameti
Andreas Unterweger
Arne Bathke
Cornelia Ferner
David Anastasiu
Elena Lloret
Eric Davis
Felix Strohmeier
Florina Piroi
Francesco Sergio Pisani
Frank Danielsen
Gabriela Viale Pereira
Giuseppe Manco
Gregor Eibl
Günther Eibl
Hans Georg Bayer
Johannes Scholz
Karl Entacher
Karl-Heinz Weidmann
Klaus Frick
Manolis Koubarakis
Maximilian Tschuchnig
Michael Gadermayr
Michael Hellwig
Peter Haber
Robert Merz
Ross King
Sebastian Hegenbart
Simon Kranzer
Steffen Finck
Thomas Lampholtshammer
Willibald Krenn

Contents

RESEARCH TRACK

German Abstracts

Evaluation of Hyperparameter-Optimization Approaches in an Industrial Federated Learning System

S. Holly, T. Hiessl, S. R. Lakani, D. Schall, C. Heitzinger and J. Kemnitz

Das Federated Learning (FL) entkoppelt das Training von Modellen, von der Notwendigkeit eines direkten Datenzugriffs und ermöglicht es Unternehmen, mit Partnern aus der Industrie zusammenzuarbeiten, um ein zufriedenstellendes Leistungsniveau zu erreichen, aber ohne sensible Geschäftsinformationen zu teilen. Die Leistung eines Algorithmus für maschinelles Lernen hängt stark von der Wahl seiner Hyperparameter ab. In einer FL-Umgebung stellt die Optimierung der Hyperparameter eine neue Herausforderung dar. In dieser Arbeit wurden die Auswirkungen verschiedener Hyperparameter-Optimierungsansätze in einem FL-System untersucht. In dem Bestreben, die Kommunikationskosten, einen kritischen Engpass in FL zu reduzieren, wurde ein lokaler Hyperparameter-Optimierungsansatz untersucht, der - im Gegensatz zu einem globalen Hyperparameter-Optimierungsansatz - jedem Client seine eigene Hyperparameter-Konfiguration erlaubt. Diese Ansätze wurden auf der Grundlage von Gridsearch und Bayesian Optimization implementiert und infolgedessen die Algorithmen mit dem MNIST-Datensatz mit einer i.i.d. (dt.: unabhängig und gleichverteilt) Partition und einem Internet of Things (IoT)-Sensordatensatz aus der Industrie mit einer nicht i.i.d. Partition konfiguriert.

Towards Robust and Transferable IIoT Sensor based Anomaly Classification using Artificial Intelligence

J. Kemnitz, T. Bierweiler, H. Grieb, S. von Dosky and D. Schall

Der zunehmende Einsatz kostengünstiger Industrial Internet of Things - Sensorplattformen in Industrieanlagen bietet große Chancen für die Erkennung von Anomalie. Die Leistung eines solchen Klassifizierungsmodells hängt stark von den verfügbaren Trainingsdaten ab. Modelle funktionieren gut, wenn diese von der gleichen Maschine stammen. Sobald die Maschine jedoch ausgetauscht, repariert oder in einer anderen Umgebung in Betrieb genommen wird, ist eine Vorhersage nicht absehbar. Aus diesem Grund wurde untersucht, ob es möglich ist, eine robuste und übertragbare Methode zur KI-basierten Anomalie-Erkennung zu entwickeln, indem verschiedene Modelle und Vorverarbeitungsschritte für Kreiselpumpen verwendet werden, die vorerst getrennt und in derselben sowie in unterschiedlichen Umgebungen wieder in Betrieb genommen werden. Außerdem wurde die Modellleistung an verschiedenen Pumpen desselben Typs - im Vergleich zu den vorhandenen Trainingsdaten - untersucht.

Data-driven Cut-off Frequency Optimization for Biomechanical Sensor Data Pre-Processing

S. Bernhart, V. Venek, C. Kranzinger, W. Kremser and A. Martínez

Bei der Vorbereitung von biomechanischen Sensordaten werden häufig Signalfilter zur Rauschunterdrückung eingesetzt, um die Leistung von Segmentierungs- und maschinellen Lernalgorithmen zu verbessern. Die Suche nach einem optimalen Wert für die Grenzfrequenz des Filters ist jedoch zeitaufwendig, da sich die Forscher auf Heuristiken und Erfahrung verlassen müssen. Daher wurde eine Methode namens FcOpt entwickelt, um automatisch eine optimale Grenzfrequenz für die Rauschfilterung in eindimensionalen biomechanischen Daten zu ermitteln. Die Methode führt eine erneute Abtastung der Eingabedaten durch und wendet drei automatische Varianten zur Bestimmung der Grenzfrequenz an, führt anschließend deren individuell vorgeschlagene Grenzfrequenzen mit einem k-means-Cluster-Algorithmus zusammen und liefert eine optimale Grenzfrequenz für die Filterung eindimensionaler Datenströme. FcOpt wird exemplarisch im Zusammenhang mit einem Algorithmus zur Segmentierung von Skischwüngen angewendet. Diese Methode wirkt der - durch hohe Abtastraten bedingten - Anfälligkeit automatisierter Verfahren für die Identifizierung von Grenzfrequenzen entgegen. FcOpt schlägt eine Cut-Off-Frequenz von 2,63 Hz, statt der ursprünglich vorgeschlagenen 3 Hz vor. Die Filterung mit der empfohlenen Grenzfrequenz weicht im Durchschnitt um 1,0 ms von der ursprünglichen zeitlichen Genauigkeit der Skischwung-Segmentierung ab, was nur 0,08% in Bezug auf die mittlere Schwungdauer entspricht. Obwohl FcOpt Heuristiken zur Bestimmung der Grenzfrequenz noch nicht vollständig ersetzen kann, ist es ein einfaches zu verwendendes Werkzeug für Forscher, welche die Signalvorverarbeitung für ihre Segmentierungsalgorithmen verbessern wollen. Es legt den Grundstein für künftige Entwicklungen auf dem Gebiet des datengesteuerten Filterdesigns.

A Low-Complexity Deep Learning Framework For Acoustic Scene Classification

L. Pham, H. Tang, A. Jalali, A. Schindler, R. King and I. McLoughlin

In diesem Beitrag wird ein Low-Complexity Deep Learning Framework für acoustic scene classification (ASC) vorgestellt. Das vorgeschlagene Framework kann in drei Hauptschritte unterteilt werden: Front-end spectrogram extraction, back-end classification, und eine Zusammenführung der vorhergesagten Wahrscheinlichkeiten. Zunächst werden Mel filter, Gammatone filter, and Constant Q Transform (CQT) eingesetzt, um rohe Audiosignale in Spektrogramme umzuwandeln, in denen sowohl Frequenzmerkmale als auch zeitliche Merkmale dargestellt werden. Drei Spektrogramme werden dann in drei faltungsneuronale Netze eingespeist, die zehn Szenen im städtischen Umfeld klassifizieren. Schließlich wird eine späte Fusion von drei vorhergesagten Wahrscheinlichkeiten, die von drei CNNs beigesteuert werden, durchgeführt, um das endgültige Klassifikationsergebnis zu erhalten. Um die Komplexität des vorgeschlagenen CNN-Netzes zu reduzieren, wurden zwei Kompressionstechniken angewendet: Modellrestriktion und dekomponierte Faltung (=model restriction and decomposed convolution). Bei den umfangreichen Experimenten, die mit den DCASE 2021 (IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events) Task 1A Development and Evaluation Datensets durchgeführt wurden, konnte ein CNN-basiertes Framework mit geringer Komplexität und 128 KB trainierbaren Parametern mit einer Klassifizierungsgenauigkeit 66,7% sowie 69,6% erreicht werden, was die DCASE-Baseline um 19,0% bzw. 24,0% verbessert.

Anomaly Detection in Medical Imaging - A Mini Review

M. E. Tschuchnig and M. Gadermayr

Die zunehmende Digitalisierung der medizinischen Bildgebung ermöglicht - auf maschinellem Lernen basierende Verbesserungen - bei der Erkennung, Visualisierung und Segmentierung von Läsionen, was die Arbeit von medizinischen Experten erleichtert. Für das überwachte maschinelle Lernen werden jedoch zuverlässige gelabelte Daten benötigt, die oft nur schwer oder gar nicht zu beschaffen sind oder zumindest zeitaufwändig und damit kostspielig aufbereitet werden müssen. Daher werden immer häufiger Methoden eingesetzt, die nur teilweise über Labels überwachtes Lernen unterstützen oder gar keine Labels (nicht überwacht) benötigen. Die Anomalie-Erkennung ist eine mögliche Vorgangsweise, bei der halbüberwachte und nicht überwachte Methoden eingesetzt werden, um Aufgaben der medizinischen Bildgebung wie Klassifizierung und Segmentierung zu bewältigen. Im eingereichten Beitrag dient ein umfangreicher Literaturüberblick hinsichtlich relevanter Arbeiten zur Anomalie-Erkennung in der medizinischen Bildgebung als Grundlage. Hiermit versucht man Anwendungen zu gruppieren, wichtige Ergebnisse hervorzuheben, Lehren daraus zu ziehen und weitere Ratschläge für die Vorgehensweise bei der Anomalie-Erkennung in der medizinischen Bildgebung zu geben. Die qualitative Analyse basierte auf Google Scholar und 4 verschiedenen Suchbegriffen, wodurch 120 verschiedene Artikel analysiert wurden. Die wichtigsten Ergebnisse zeigen, dass die derzeitige Forschung hauptsächlich bestrebt ist, die Verringerung des Bedarfs an gelabelten Daten, bei annähernd gleichbleibender Qualität der Ergebnisse, zu erreichen. Die erfolgreiche und umfangreiche Forschung auf dem Gebiet des Hirn-MRT zeigt auch das Potenzial für Anwendungen in anderen Bereichen wie OCT und Thorax-Röntgen.

Deep Learning Frameworks Applied For Audio-Visual Scene Classification

L. Pham, A. Schindler, M. Schütz, J. Lampert, S. Schlarb and R. King

In diesem Beitrag werden Deep-Learning-Frameworks für die audiovisuelle Szenenklassifikation (SC) vorgestellt und zeigen auf, wie einzelne visuelle und akustische Merkmale sowie deren Kombination die SC-Leistung beeinflussen. Die umfangreichen Experimente wurden mit den DCASE 2021 (IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events) Task 1B Entwicklungs- und Bewertungsdatensätze durchgeführt. Die Ergebnisse mit dem Entwicklungsdatensatz erzielen die beste Klassifizierungsgenauigkeiten von 82,2%, 91,1% und 93,9% mit jeweils nur Audioeingabe, nur visueller Eingabe sowie Audio- und auch visueller Eingabe. Die höchste Klassifizierungsgenauigkeit von 93,9%, die von einem Ensemble aus audio- und visuell-basierten Frameworks erzielt wurde, zeigt eine Verbesserung von 16,5% im Vergleich zur DCASE 2021 Baseline. Das beste Ergebnis im Evaluationsdatensatz beträgt 91,5% und übertrifft damit die DCASE-Baseline von 77,1%.

Toward Applying the IEC 62443 in the UAS for Secure Civil Applications

A. M. Shaaban, O. Jung and M. A. F. Millan

Die wachsende Nachfrage nach Drohnen für zivile Anwendungen wird in der Regel mit kommerziellen Standardgeräten bedient. Diese können zwar immer an die Bedürfnisse des Endnutzers angepasst werden, erfüllen aber nicht alle kritischen Aspekte wie Leistung, Effizienz oder Sicherheit. Die Cybersicherheit ist eines der kritischen Themen bei unbemannten Luftfahrtsystemen (engl.: Unmanned Aircraft Systems UAS), bei denen Cyberangriffe auf dieses System zu zahlreichen negativen Folgen führen können. Im vorliegenden Paper wird das Thema Cybersicherheit behandelt, indem eine Reihe strategischer Maßnahmen vorgestellt werden, um einen vollständigen Entwicklungsprozess für die Erstellung sicherer UAV-Anwendungen (Unmanned Aerial Vehicle) zu definieren. Neben der Implementierung des Sicherheitsstandards IEC 62443 in UAS wird ein umfassender Katalog mit Bedrohungen, Komponenten und kritischen Werten für UAVs erstellt. In der Folge wird das sogenannte ThreatGet-Tool eingesetzt, um automatisch relevante Bedrohungen zu identifizieren und zu bestimmen bzw. die Risikostufe im Zusammenhang mit einer UAS-Fallstudie abzuschätzen. Die Ergebnisse von ThreatGet werden verwendet, um einen Überblick über ein Mapping-Verfahren zwischen Bedrohungen und Sicherheitsanforderungen zu geben. Diese Strategie zielt vornehmlich darauf ab, eine Reihe von Sicherheitsanforderungen zu ermitteln, um potenziellen Bedrohungen zu adressieren und kritische Werte in UAS zu schützen.

IAIDO: A Framework for Implementing Integrity-Aware Intelligent Data Objects

E. Davis

Die zunehmende Abhängigkeit von automatisiertem Denken, maschinellem Lernen und maschinengestützter Entscheidungsfindung hat zu ernsthaften Schwachstellen im Bereich der Datenintegrität geführt. Der vertrauenswürdige und zuverlässige Betrieb von datengesteuerten Systemen der nächsten Generation und der Infrastruktur, die diese Daten verwaltet, erfordert wirksame und skalierbare Lösungen für die wachsende Gefahr von Fehlern aufgrund von Datenintegrität. In diesem Beitrag wird das Konzept der Datenintegrität diskutiert, die Bedrohungen für die Datenintegrität skizziert und der Begriff der integritätsbewussten Datenobjekte vorgestellt, welche die Konzepte des Polymorphismus, der Subsumption, der Komposition, der Assoziation und der Aggregation nutzen, um ein System zur Verbesserung der Datenintegrität für große Datensätze mit gemeinsamer Herkunft, Repräsentationen und Typen aufzubauen. Der Begriff dieser Datenobjekte wird dahingehend erweitert, indem Intelligenz in Form von erlernten Einschränkungen, Regeln und Klassifikatoren hinzugefügt werden, die von Datenobjekten geerbt werden, um die Toleranz gegenüber Datenintegritätsfehlern zu verbessern. Diese integritätsbewussten intelligenten Datenobjekte werden als IAIDO-Framework implementiert. Der neuartige Ansatz wird anhand von realen Daten zu Ernährungsinformationen dargestellt, indem Beispiele für reale Datenintegritätsfehler in der National Nutrient Data Base for Standard Reference Release 28 des USDA und in Crowd-Sourced-Daten verwendet werden. Schließlich werden die hohen Raten von Datenintegritätsfehlern in Crowd-Sourced-Daten gezeigt, wobei fast 27% der Daten eine oder mehrere SMT-basierte Einschränkung/en nicht erfüllen. Ähnlich verhält es sich mit den vom USDA veröffentlichten Daten: fast 10% der Daten sind nicht konform und weisen Fehler in der Datenintegrität auf.

Reducing Operator Overload with Context-Sensitive Event Clustering

M. Basalla, J. Schneider and J. vom Brocke

Die Betreiber komplexer, vernetzter Systeme sind ständig mit einer großen Anzahl von Fehlerereignissen konfrontiert, deren Behebung viel Zeit in Anspruch nimmt. Ereignisse in einer Netzkomponente können eine Reihe weiterer Ereignisse in anderen Komponenten auslösen, was bei vielen miteinander verknüpften Sequenzen zu einer großen Anzahl von Fehlermeldungen führt. Betreiber versuchen in der Regel, die Grundursache einer Folge von Ereignissen, die auf ein Problem hinweisen, zu identifizieren und zu verstehen, da die Behebung der Grundursache in der Regel die Folgeprobleme aufzeigt. Auf der Grundlage eines realen Datensatzes werden zwei Techniken vorgestellt, mit denen die Anzahl der Ereignisse und Fehlerprotokolle reduziert werden kann, ohne die Grundursache zu vernachlässigen. Eine Technik nutzt vorhandene Process-Mining-Tools in Kombination mit manueller Analyse. Das andere Verfahren beruht auf der Berechnung kontextsensitiver Einbettungen, ähnlich den Worteinbettungen bei der Verarbeitung natürlicher Sprache. Die Einbettungen werden zum Clustern von Ereignistypen verwendet, um das gemeinsame Auftreten und die Kausalität zwischen ihnen zu ermitteln. Obwohl beide Techniken ihre Stärken und Schwächen haben, reduzieren sie die Anzahl möglicher Ereignisse erheblich, während sie gleichzeitig die Bedingungen für die Kausalität durchsetzen.

Dynamic Review-based Recommenders

K. Cvejoski, R. J. Sánchez, C. Bauckhage and C. Ojeda

Im gleichen Ausmaß wie sich Präferenzen von Nutzern im Laufe der Zeit ändern, spiegeln die Rezensionen von Artikeln diese Änderungen von Präferenzen wider. Kurz gesagt, wenn man das Wissen über den Inhalt von Rezensionen in Empfehlungssysteme einbeziehen will, endet man auf natürliche Weise bei dynamischen Textmodellen. In der vorliegenden Arbeit wurde die bekannte Stärke von Rezensionen eingesetzt, um die Vorhersage von Bewertungen zu verbessern, und zwar auf eine Art und Weise, die (i) die Kausalität der Rezensionserstellung respektiert und (ii) in einer bi-direktionalen Weise die Fähigkeit von Bewertungen einbezieht, sprachliche Rezensionsmodelle zu informieren und im umgekehrten Falle Bewertungen vorherzusagen. Darüber hinaus sind die Darstellungen zeitintervallabhängig und liefern somit eine zeitkontinuierliche Darstellung der Dynamik. Die Experimente wurden mit realen Datensätzen durchgeführt und zeigen, dass die Methodik in der Lage ist, mehrere State-of-the-Art-Modelle zu übertreffen.

Full Papers - Peer Reviewed

Predictive Maintenance and Hyperparameter Optimization in the Industrial Setting

Evaluation of Hyperparameter-Optimization Approaches in an Industrial Federated Learning System

Stephanie Holly[*†], Thomas Hiessl[*], Safoura Rezapour Lakani[*],
Daniel Schall[*], Clemens Heitzinger[†] and Jana Kemnitz[*]
[*]Siemens Technology, 1210 Vienna, Austria
[†]TU Wien, 1040 Vienna, Austria
{e11703485@student, clemens.heitzinger}@tuwien.ac.at,
{hiessl.thomas, safoura.rezapour_lakani, daniel.schall, jana.kemnitz}@siemens.com

***Abstract*—Federated Learning (FL) decouples model training from the need for direct access to the data and allows organizations to collaborate with industry partners to reach a satisfying level of performance without sharing vulnerable business information. The performance of a machine learning algorithm is highly sensitive to the choice of its hyperparameters. In an FL setting, hyperparameter optimization poses new challenges. In this work, we investigated the impact of different hyperparameter optimization approaches in an FL system. In an effort to reduce communication costs, a critical bottleneck in FL, we investigated a local hyperparameter optimization approach that – in contrast to a global hyperparameter optimization approach – allows every client to have its own hyperparameter configuration. We implemented these approaches based on grid search and Bayesian optimization and evaluated the algorithms on the MNIST data set using an i.i.d. partition and on an Internet of Things (IoT) sensor based industrial data set using a non-i.i.d. partition.**

***Index Terms*—Industrial federated learning, Optimization approaches, Hyperparameter optimization**

I. Introduction

The performance of a machine learning algorithm is highly sensitive to the choice of its hyperparameters. Therefore, hyperparameter selection is a crucial task in the optimization of knowledge-aggregation algorithms. Federated Learning (FL) is a recent machine learning approach which aggregates machine learning model parameters between devices (henceforth clients) without sharing their data. The aggregation is coordinated by a server. Industrial Federated Learning (IFL) is a modified approach of FL in an industrial context [1]. In an FL setting, hyperparameter optimization poses new challenges and is a major open research area [2]. In this work, we investigate the impact of different hyperparameter optimization approaches in an IFL system. We believe that the data distribution influences the choice of the best hyperparameter configuration and suggest that the best hyperparameter configuration for a client might differ from another client based on individual data properties. Therefore, we investigate a local hyperparameter optimization approach that – in contrast to a global hyperparameter optimization approach – allows every client to have its own hyperparameter configuration. The local approach allows us to optimize hyperparameters prior to the federation process reducing communication costs.

Communication is considered a critical bottleneck in FL [3]. Clients are usually limited in terms of communication bandwidth enhancing the importance of reducing the number of communication rounds or using compressed communication schemes for the model updates to the central server [3]. Dai et al. [4] introduced *Federated Bayesian Optimization* (FBO) extending Bayesian optimization to the FL setting. However, until now, there is no research on the impact of global and local hyperparameter optimization in FL. Therefore, we compare a local hyperparameter optimization approach to a global hyperparameter optimization approach, optimizing hyperparameters in the federation process.

The aim of this work is to i) analyze challenges and formal requirements in FL, and in particular in IFL, ii) to evaluate the performance of an Internet of Things (IoT) sensor based classification task in an IFL system, iii) to investigate a communication efficient hyperparameter optimization approach, and iv) to compare different hyperparameter optimization algorithms. Therefore, we want to answer the following questions.

Q1: Does FL work for an IoT sensor based anomaly classification task on industrial assets with non-identically distributed data in an IFL system with a cohort strategy?

Q2: Can we assume that the global and local hyperparameter optimization approach deliver the same hyperparameter configuration in an i.i.d. FL setting?

Q3: Can we reduce communication costs in the hyperparameter optimization of a non-i.i.d. classification task in context of FL by optimizing a hyperparameter locally prior to the federation process?

Q4: Does Bayesian optimization outperform grid search, both in a global and local approach of a non-i.i.d. IoT sensor based classification task?

II. Algorithmic Challenges and Formal Requirements for Industrial Assets

In FL, new algorithmic challenges arise that differentiate the corresponding optimization problem from a distributed

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_1

optimization problem. In distributed learning settings, major assumptions regarding the training data are made which usually fail to hold in an FL setting [5]. Moreover, non-i.i.d. data, limited communication, and limited and unreliable client availability pose further challenges for optimization problems in FL [2]. Kairouz et al. [2] considered the need for addressing these challenges as a major difference to distributed optimization problems. The optimization problem in FL is therefore referred to as federated optimization emphasizing the difference to distributed optimization [5]. In an IFL setting, additional challenges regarding industrial aspects arise [1]. In this section, we want to formulate the federated optimization problem and discuss the algorithmic challenges of FL in general, and in particular of IFL.

A. Problem Formulation

We consider a supervised learning task with features x in a sample space $\mathcal{X}$ and labels y in a label space $\mathcal{Y}$. We assume that we have K available clients, $K \in \mathbb{N}_{\geq 2}$, with

$$D_k := D_{\mathcal{X},k} \times D_{\mathcal{Y},k} \subseteq \mathcal{X} \times \mathcal{Y}$$

denoting the data set of client k and $n_k := |D_k|$ denoting the cardinality of the client's data set. Let $\mathcal{Q}$ denote the distribution over all clients, and let $\mathcal{P}_k$ denote the data distribution of client k. We can then access a specific data point by first sampling a client $k \sim \mathcal{Q}$ and then sampling a data point $(x, y) \sim \mathcal{P}_k$ [2]. Then, the local objective function is

$$F_k(w) := \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}_k}[f(x, y, w)], \tag{1}$$

where $w \in \mathbb{R}^d$ represents the parameters of the machine learning model and $f(x, y, w)$ represents the loss of the prediction on sample (x, y) for the given parameters w. Typically, we wish to minimize

$$F(w) := \frac{1}{K}\sum_{k=1}^{K} F_k(w). \tag{2}$$

B. Federated Learning

One of the major challenges concerns data heterogeneity. In general, we cannot assume that the data is identically distributed over the clients, that is $\mathcal{P}_k = \mathcal{P}_l$ for all k and l. Therefore, F_k might be an arbitrarily bad approximation of F [5].

In the following, we want to analyze different non-identically distributed settings as demonstrated by Hsieh et al. [6] assuming that we have an IoT sensor based anomaly classification task in an industrial context. Given the distribution $\mathcal{P}_k$, let $P^k_{\mathcal{X},\mathcal{Y}}$ denote the bivariate probability function, let $P^k_{\mathcal{X}}$ and $P^k_{\mathcal{Y}}$ denote the marginal probability function respectively. Using the conditional probability function $P^k_{\mathcal{Y}|\mathcal{X}}$ and $P^k_{\mathcal{X}|\mathcal{Y}}$, we can now rewrite the bivariate probability function as

$$P^k_{\mathcal{X},\mathcal{Y}}(x, y) = P^k_{\mathcal{Y}|\mathcal{X}}(y|x)P^k_{\mathcal{X}}(x) = P^k_{\mathcal{X}|\mathcal{Y}}(x|y)P^k_{\mathcal{Y}}(y) \tag{3}$$

for $(x, y) \in \mathcal{X} \times \mathcal{Y}$. This allows us to characterize different settings of non-identically distributed data:

Feature distribution skew: We assume that $P^k_{\mathcal{Y}|\mathcal{X}} = P^l_{\mathcal{Y}|\mathcal{X}}$ for all k, l, but $P^k_{\mathcal{X}} \neq P^l_{\mathcal{X}}$ for some k, l. Clients that have the same anomaly classes might still have differences in the measurements due to variations in sensor and machine type.

Label distribution skew: We assume that $P^k_{\mathcal{X}|\mathcal{Y}} = P^l_{\mathcal{X}|\mathcal{Y}}$ for all k, l, but $P^k_{\mathcal{Y}} \neq P^l_{\mathcal{Y}}$ for some k, l. The distribution of labels might vary across clients as clients might experience different anomaly classes.

Same label, different features: We assume that $P^k_{\mathcal{Y}} = P^l_{\mathcal{Y}}$ for all k, l, but $P^k_{\mathcal{X}|\mathcal{Y}} \neq P^l_{\mathcal{X}|\mathcal{Y}}$ for some k, l. The same anomaly class can have significantly different features for different clients due to variations in machine type, operational- and environmental conditions.

Same features, different label: We assume that $P^k_{\mathcal{X}} = P^l_{\mathcal{X}}$ for all k and l, but $P^k_{\mathcal{Y}|\mathcal{X}} \neq P^l_{\mathcal{Y}|\mathcal{X}}$ for some k, l. The same features can have different labels due to operational- and environmental conditions, variation in manufacturing, maintenance et cetera.

Quantity skew: We cannot assume that different clients hold the same amount of data, that is $n_k = n_l$ for all k, l. Some clients will generate more data than others.

In real-world problems, we expect to find a mixture of these non-identically distributed settings. In FL, heterogeneity does not exclusively refer to a non-identical data distribution, but also addresses violations of independence assumptions on the distribution $\mathcal{Q}$ [2]. Due to limited, slow and unreliable communication on a client, the availability of a client is not guaranteed for all communication rounds. Communication is considered a critical bottleneck in FL [3]. In each communication round, the participating clients send a full model update w back to the central server for aggregation. In a typical FL setting, however, the clients are usually limited in terms of communication bandwidth [3]. Consequently, it is crucial to minimize the communication costs.

C. Industrial Federated Learning

In an industrial setting, FL experiences challenges that specifically occur in an industrial context. Industrial assets have access to a wealth of data suitable for machine learning models, however, the data on an individual asset is typically limited and private in nature. In addition to sharing the model within the company, it can also be shared with an external industry partner [1]. FL leaves possibly critical business information distributed on the individual client (or within the company). However, Zhao et al. [7] proved that heterogeneity, in particular, a highly skewed label distribution, significantly reduces the accuracy of the aggregated model in FL. In an industrial context, we expect to find heterogeneous clients due to varying environmental and operational conditions on different assets. Therefore, Hiessl et al. [1] introduced a modified approach of FL in an industrial context and termed it *Industrial Federated Learning* (IFL). IFL does not allow arbitrary knowledge exchange between clients. Instead, the knowledge exchange only takes place between clients that have sufficiently similar data. Hiessl et al. [1] refer to this set of clients as a *cohort*. We expect the federated learning approach in a cohort to approximate the corresponding central learning approach.

III. Hyperparameter Optimization Approaches in an IFL System

In an FL setting, hyperparameter optimization poses new challenges and is a major open research area [2]. The performance of a machine learning model is linked to the amount of communication [8]. In an effort to reduce communication costs, a critical bottleneck in FL [3], we investigated a communication efficient hyperparameter optimization approach, a local hyperparameter optimization approach that allows us to optimize hyperparameters prior to the federation process. Kairouz et al. [2] introduced the idea of a separate optimization of hyperparameters and suggest a different hyperparameter choice for dealing with non-i.i.d. data.

Dai et al. [4] investigated a communication efficient local hyperparameter optimization approach and introduced Federated Bayesian Optimization (FBO) extending Bayesian optimization to the FL setting. In FBO, every client locally uses Bayesian optimization to find the optimal hyperparameter configuration. Additionally, each client is allowed to request for information from other clients. Dai et al. [4] proved a convergence guarantee for this algorithm and its robustness against heterogeneity. However, until now, there is no research on the impact of global and local hyperparameter optimization.

In the LocalHPO algorithm 1, we perform local hyperparameter optimization. We optimize the hyperparameter configuration λ^k for each client k. In the GlobalHPO algorithm 2, we perform global hyperparameter optimization. We optimize the hyperparameter configuration λ in the federation process. The LocalOptimization method in the LocalHPO algorithm 1 and the GlobalOptimization method in the GlobalHPO algorithm 2 can be based on any hyperparameter optimization algorithm.

Algorithm 1: LocalHPO

```
Server executes:
initialize $w_0$
for each client $k = 1, \dots, K$ do
    $\lambda^k :=$ LocalOptimization($k, w_0$)
end
return $(\lambda^k)_{k=1}^K$
```

Algorithm 2: GlobalHPO

```
Server executes:
$\lambda :=$ GlobalOptimization()
return $\lambda$
```

We want to differentiate between a global hyperparameter λ_i whose value is constant for all clients and a local hyperparameter λ_i^k whose value depends on a client k. Here, λ_i^k denotes the hyperparameter λ_i on client k. We notice that this differentiation is only relevant for settings with non-i.i.d. data. In an i.i.d. setting, we assume that a hyperparameter configuration that works for one client also works for another client. In our experiments, we verified this assumption for a proxy data set.

IV. Data, Algorithms and Experiments

In the next section, we want to make our benchmark design explicit and present our experimental setup. We will present the machine learning tasks including the data partition of the training data, the machine learning models, the optimization algorithms and our experiments. We considered an image classification task on a data set, the MNIST data set of handwritten digits, and an IoT sensor based anomaly classification task on industrial assets.

A. Data

In order to test the IFL system on the MNIST data set, we still need to specify on how to distribute the data over artificially designed clients. To systematically evaluate the effectiveness of the IFL system, we simulated an i.i.d. data distribution. This refers to shuffling the data and partitioning the data into 10 clients, each receiving 6 000 examples. Following the approach of McMahan et al. [5], we applied a convolutional neural network with the following settings: 2 convolutional layers with 32 and 64 filters of size 5×5 and a ReLu activation function, each followed by a max pooling layer of size 2×2, a dense layer with 512 neurons and a ReLu activation function, a dense layer with 10 neurons and a softmax activation function.

The industrial task concerns IoT sensor based anomaly classification on industrial assets. The data was acquired with the SITRANS multi sensor, specifically developed for industrial applications and its requirements [9]. We considered multiple centrifugal pumps with sensors placed at different positions, in different directions to record three axis vibrational data in a frequency of 6644 Hz. Per minute, 512 samples were collected. We operated the pumps under 6 varying conditions, including 3 healthy states and 3 anomalous states. A client is either assigned data of an asset in a measurement, or data of several assets in a measurement ensuring that each client sees all operating conditions. However, since in the process of measurement, the assets were completely dismantled and rebuilt, we consider the data to be non-i.i.d. regarding its feature distribution. We applied an artificial neural network with the following settings: a dense layer with 64 neurons and a ReLu activation function, a dropout layer with a dropout rate of 0.4, a dense layer with 6 neurons and a ReLu activation function, a dropout layer with a dropout rate of 0.4, and a softmax activation function. We remapped the features using the Kabsch algorithm [10], applied a sliding window, extracted the Melfrequency cepstral coefficients, applied the synthetic minority oversampling technique [10], and normalized the resulting features.

B. Algorithms

Our evaluations include the Federated Averaging (FedAvg) algorithm according to McMahan et al. [5], and the hyperparameter optimization approaches LocalHPO 1 and GlobalHPO 2. We implemented these approaches based on grid search and Bayesian optimization. In this section, we give their pseudocode. We searched for the learning rate η with fixed fraction of participating clients C, number of communication rounds R, number of local epochs E, and mini-batch size B.

In algorithm 3, we give the pseudocode of the LocalOptimization method in LocalHPO 1 based on the grid search algorithm with a fixed grid G. We iterate through the grid G, train the model on the training data of client k based on the ClientUpdate method used in the FedAvg algorithm [5] with the learning rate η as an additional argument, and validate the performance of the model w_η on the validation data $\mathcal{D}^k_{\text{valid}}$ of client k. Finally, the learning rate that yields the highest accuracy A_η on the validation data is selected. Here, w_η denotes the resulting model trained on the training data with learning rate η and $A(\mathcal{D}^k_{\text{valid}}, w_\eta)$ denotes the accuracy of the model tested on the validation data $\mathcal{D}^k_{\text{valid}}$ of client k.

Algorithm 3: Local Grid Search

```
LocalOptimization($k, w_0$):
for each learning rate $\eta \in G$ do
    $w_\eta :=$ ClientUpdate($k, w_0, \eta$)
    $A_\eta := A(\mathcal{D}^k_{\text{valid}}, w_\eta)$
end
$\eta^*_k := \arg\max_{\eta \in G} A_\eta$
return $\eta^*_k$
```

Algorithm 4: Global Grid Search

```
GlobalOptimization():
for each learning rate $\eta \in G$ do
    $w_\eta :=$ FederatedAveraging($\eta$)
    for each client $k = 1, \dots, K$ do
        $A^k_\eta := A(\mathcal{D}^k_{\text{valid}}, w_\eta)$
    end
    $A_\eta := \frac{1}{K} \sum_{k=1}^{K} A^k_\eta$
end
$\eta^* := \arg\max_{\eta \in G} A_\eta$
return $\eta^*$
```

In algorithm 4, we give the pseudocode of the GlobalOptimization method in GlobalHPO 2 based on the grid search algorithm with a fixed grid G. We iterate through the grid, perform the FedAvg algorithm [5] with the learning rate η as an additional argument, validate the performance of the model w_η on the validation data $\mathcal{D}^k_{\text{valid}}$ for all clients k and compute the average accuracy of all clients. Finally, the learning rate that yields the highest average accuracy A_η is selected.

In algorithm 5, we give the pseudocode of the LocalOptimization method in LocalHPO 1 based on Bayesian optimization. The objective function f takes the learning rate η as an argument, trains the model on the training data of client k based on the ClientUpdate method used in the FedAvg algorithm [5] with the learning rate η as an additional argument, validates the performance of the model w on the validation data $\mathcal{D}^k_{\text{valid}}$ of client k, and returns the resulting accuracy. We initialize a Gaussian process GP for the objective function f with n_{init} sample points. Then, we find the next sample point $\eta_{n_{\text{init}}+i}$ by maximizing the acquisition function, evaluate $f(\eta_{n_{\text{init}}+i})$, and update the Gaussian process GP. Finally, we select the learning rate η^* that yields the highest accuracy. We repeat this for n_{iter} iterations.

In algorithm 6, we give the pseudocode of the GlobalOptimization method in GlobalHPO 2 based on Bayesian optimization. The objective function f takes the learning rate η as an argument, performs the FedAvg algorithm [5] with the learning rate η as an additional argument, validates the performance of the model w on the validation data $\mathcal{D}^k_{\text{valid}}$ for all clients k, computes the average accuracy of all clients and returns the resulting accuracy. We initialize a Gaussian process GP for the objective function f with n_{init} sample points. Then, we find the next sample point $\eta_{n_{\text{init}}+i}$ by maximizing the acquisition function, evaluate $f(\eta_{n_{\text{init}}+i})$, and update the Gaussian process GP. Finally, we select the learning rate η^* that yields the highest average accuracy. We repeat this for n_{iter} iterations.

Algorithm 5: Local Bayesian Optimization

```
LocalOptimization($k, w_0$):
initialize a Gaussian process $GP$ for $f$
evaluate $f$ at $n_{\text{init}}$ initial points
for $i = 1, \dots, n_{\text{iter}}$ do
    find sample point $\eta_{n_{\text{init}}+i}$ that maximizes acquisition
      function
    evaluate objective function $f$ at $\eta_{n_{\text{init}}+i}$
    update the Gaussian process $GP$
end
$\eta^* := \arg\max_{i=1,\dots,n_{\text{init}}+n_{\text{iter}}} f(\eta_i)$
return $\eta^*$
objective function:
$f(\eta)$:
$w :=$ ClientUpdate($k, w_0, \eta$)
$A := A(\mathcal{D}^k_{\text{valid}}, w)$
return $A$
```

C. Experiments

In order to systematically investigate the impact of global and local hyperparameter optimization, we compared the global and local hyperparameter optimization approach in an i.i.d. setting, the MNIST machine learning task, as well as in a non-i.i.d. setting, the industrial task. Therefore, we implemented the global and local optimization approach based on grid search with a grid $G := [0.0001, 0.001, 0.01, 0.1]$, and based on Bayesian optimization with the widely used squared exponential kernel and the upper confidence bound acquisition function. We searched for the learning rate η with fixed R, C, E and B.

In order to evaluate the global and local optimization approaches in a direct comparison, we chose the number of epochs E in the local optimization approach as $E = E_{\text{global}}R$, where E_{global} is the number of epochs in the global

Algorithm 6: Global Bayesian Optimization

```
GlobalOptimization():
initialize a Gaussian process $GP$ for $f$
evaluate $f$ at $n_{\text{init}}$ initial points
for $i = 1, \ldots, n_{\text{iter}}$ do
    find sample point $\eta_{n_{\text{init}}+i}$ that maximizes acquisition function
    evaluate objective function $f$ at $\eta_{n_{\text{init}}+i}$
    update the Gaussian process $GP$
end
$\eta^* := \arg\max_{i=1,\ldots,n_{\text{init}}+n_{\text{iter}}} f(\eta_i)$
return $\eta^*$
objective function:
$f(\eta)$:
$w :=$ FederatedAveraging$(\eta)$
for each client $k = 1, \ldots, K$ do
    $A^k := A(\mathcal{D}^k_{\text{valid}}, w)$
end
$A := \frac{1}{K}\sum_{k=1}^{K} A^k$
return A
```

optimization approach and R is the number of communication rounds. In the global optimization task, we set $R := 10$, $C := 1$, $E := 1$ and $B := 128$ for the MNIST data, and $R := 10$, $C := 1$, $E := 5$ and $B := 128$ for the industrial data. In the local optimization task, we set $E := 10$ and $B := 128$ for the MNIST data, and $E := 50$ and $B := 128$ for the industrial data. For the evaluation of the global hyperparameter optimization approach, we optimized the learning rate using the global approach, trained the federated model with a global learning rate, and tested the resulting federated model on the cohort test data. Then, we optimized the learning rate using the local approach, trained the federated model with local individual learning rates for each client in the cohort, and tested the resulting federated model on the cohort test data.

V. Experimental Results

Following the approach of Hiessl et al. [1], we demonstrated the effectiveness of the IFL System for the industrial task and showed that the IFL approach performs better than the individual learning approach and approximates the central learning approach. Fig. 1 shows the test accuracy on the central cohort test data for each client, for i) a model trained on the individual training data of the client (individual learning), ii) a central model trained on the collected training data of all clients in the cohort (central learning), and iii) the federated model trained in the cohort.

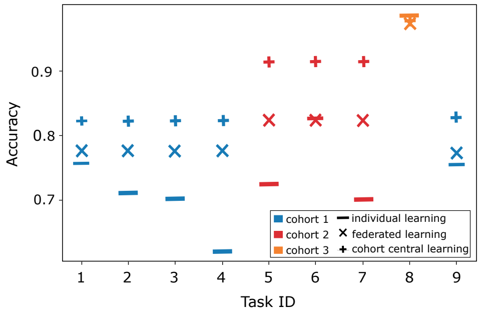


Fig. 1. Comparison of individual learning, central learning, and federated learning on the industrial data set for all clients (Task ID).

Fig. 2 a) shows the results for the MNIST data. The optimization approaches are based on the grid search algorithm. For the training posterior to the optimization, we set $R := 10$, $C := 1$, $E := 1$, and $B := 128$ in the IFL system. The color indicates the optimized learning rate on the corresponding client. Since the MNIST data is i.i.d., there is only one cohort and all clients have the same federated model and thus the same test accuracy. Our results show that the grid search algorithm selected 10^{-3} in the local optimization of the learning rate on each client. According to our expectation, the global optimization approach yielded the same learning rate.

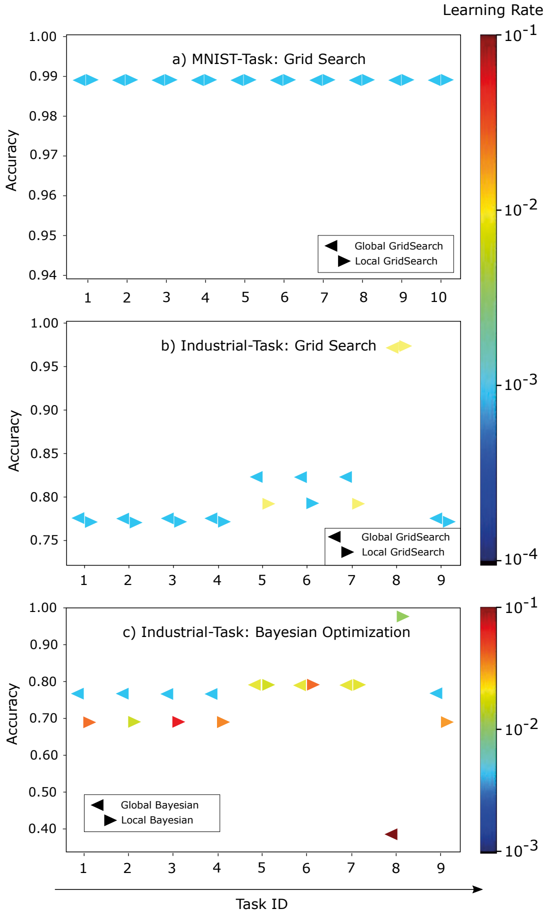


Fig. 2. Comparison of the optimization approaches based on a) grid search for the MNIST task, b) grid search for the industrial task, and c) Bayesian optimization for the industrial task for all clients (Task ID).

For the industrial task, we evaluated the global and local optimization approach based on grid search and Bayesian optimization. For the training posterior to the optimization, we set $R := 20$, $C := 1$, $E := 5$, and $B := 128$ in the IFL system. Fig. 2 b) shows the results for the industrial data with the optimization approaches based on the grid search algorithm. The results show that, in all cohorts, the global approach yielded an equal or larger accuracy than the local approach.

Fig. 2 c) shows the results for the industrial data with the optimization approaches based on the Bayesian algorithm. Note that the search space of the learning rate was $[10^{-4}, 10^{-1}]$ in the optimization while the scale in the plot starts from 10^{-3}. The results show that the global approach yielded a larger accuracy than the local approach in cohort 0 and cohort 1.

The local Bayesian approach yielded different learning rates, see Fig. 2 c), on clients with no difference in data, that is, the same number of samples, the same class distribution, and the same measurement protocol. However, the local grid search approach yielded the same learning rate as the global grid search approach, see Fig. 2 b). Therefore, we suggest that the reason lies in the implementation of the Bayesian optimization approach and a not sufficiently large number of iterations to guarantee convergence.

In order to compare the optimization approaches for the industrial task, we performed a paired t-test regarding the test accuracy to determine the statistical significance, see table I. We observe that the global optimization approach is significantly better than the local approach, both for the grid search approach ($p = 0.028$) and for the Bayesian approach ($p = 0.012$). Furthermore, the results show that the grid search approach is significantly better than the Bayesian approach, both for the global approach ($p = 0.004$) and for the local approach ($p = 0.008$). Note that we considered cohort 2 an outlier and excluded this cohort from our calculations. Cohort 2 only consists of client 8, a client whose data was not generated according to the standard measurement protocol. Without outlier removal, the global grid search approach is still significantly better than the local grid search approach ($p = 0.032$), and the local grid search approach is significantly better than the local Bayesian approach ($p = 0.010$). However, there is no significant difference in the global Bayesian approach vs. the local Bayesian approach ($p = 0.755$) and in the global grid search approach vs. the global Bayesian approach ($p = 0.230$).

TABLE I
TEST ACCURACY OF FEDERATED MODEL ON CENTRAL COHORT TEST DATA OF INDUSTRIAL TASK POSTERIOR TO CORRESPONDING OPTIMIZATION APPROACH AND TRAINING

client	global grid	local grid	global Bayesian	local Bayesian
1	**0.7756**	0.7720	0.7659	0.6897
2	**0.7756**	0.7720	0.7659	0.6897
3	**0.7756**	0.7720	0.7659	0.6897
4	**0.7756**	0.7720	0.7659	0.6897
5	**0.8230**	0.7921	0.7882	0.7889
6	**0.8230**	0.7921	0.7882	0.7889
7	**0.8230**	0.7921	0.7882	0.7889
8	0.9740	**0.9749**	0.3867	0.9736
9	**0.7756**	0.7720	0.7659	0.6897

VI. Conclusion and Future Work

The results show that the federated learning approach approximates the central learning approach, while outperforming individual learning of the clients. In this work, we investigated the impact of global and local optimization approaches in an IFL System based on a proxy data set and a real-world problem. In our experiments on the industrial data, local optimization yielded different learning rates on different clients in a cohort. However, the results show that a globally optimized learning rate, and thus, a global learning rate for all clients in a cohort improves the performance of the resulting federated model. Therefore, we conclude that the global optimization approach outperforms the local optimization approach resulting in a communication-performance trade-off in the hyperparameter optimization in FL. In our experiments on the proxy data set, however, the local approach achieved the same performance as the global approach.

A limitation of our study is that we only considered one hyperparameter in our optimization task. Hence it would be interesting to explore whether we can confirm these observations for a hyperparameter configuration of more hyperparameters. The results show that the grid search approaches outperform the Bayesian approaches, both globally and locally. However, we suggest a convergence analysis for the Bayesian approach.

References

[1] T. Hiessl, S. Rezapour Lakani, J. Kemnitz, D. Schall, and S. Schulte, “Cohort – based federated learning services for industrial collaboration on the edge,” *TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.14852361.v1*, 2021.

[2] P. Kairouz, H. B. McMahan, and et al., “Advances and open problems in federated learning,” *Foundations and Trends® in Machine Learning*, vol. 14, no. 1, 2021.

[3] J. Konečný, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, “Federated learning: Strategies for improving communication efficiency,” *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[4] Z. Dai, B. K. H. Low, and P. Jaillet, “Federated Bayesian optimization via Thompson sampling,” *Advances in Neural Information Processing Systems 33*, 2020.

[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data,” *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 54, pp. 1273–1282, 2017.

[6] K. Hsieh, A. Phanishayee, O. Mutlu, and P. B. Gibbons, “The non-iid data quagmire of decentralized machine learning,” *International Conference on Machine Learning (ICML)*, pp. 4387–4398, 2020.

[7] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, “Federated learning with non-iid data,” *arXiv: 1806.00582*, 2018.

[8] A. Nilsson, S. Smith, G. Ulm, E. Gustavsson, and M. Jirstrand, “A performance evaluation of federated learning algorithms,” *Proceedings of the Second Workshop on Distributed Infrastructures for Deep Learning (DIDL)*, pp. 1–8, 2018.

[9] T. Bierweiler, H. Grieb, S. von Dosky, and M. Hartl, “Smart sensing environment – use cases and system for plant specific monitoring and optimization,” *Automation 2019*, pp. 155–158, 2019.

[10] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling technique,” *J. Artif. Intell. Res*, vol. 16, pp. 321–357, 2002.

[11] F. L. Markley, “Attitude determination using vector observation: A fast optimal matrix algorithm,” *J. Astronaut. Sci.*, vol. 41, no. 2, pp. 261–280, 1993.

Towards Robust and Transferable IIoT Sensor based Anomaly Classification using Artificial Intelligence

Jana Kemnitz[*], Thomas Bierweiler[†], Herbert Grieb[†], Stefan von Dosky[†] and Daniel Schall[*]
[*]Siemens Technology, 1210 Vienna, Austria
[†]Siemens Digital Industries, 76187 Karlsruhe, Germany
{jana.kemnitz, thomas.bierweiler, herbert.grieb, stefan.von_dosky, daniel.schall}@siemens.com

***Abstract*—The increasing deployment of low-cost industrial IoT (IIoT) sensor platforms on industrial assets enables great opportunities for anomaly classification in industrial plants. The performance of such a classification model depends highly on the available training data. Models perform well when the training data comes from the same machine. However, as soon as the machine is changed, repaired, or put into operation in a different environment, the prediction often fails. For this reason, we investigate whether it is feasible to have a robust and transferable method for AI based anomaly classification using different models and pre-processing steps on centrifugal pumps which are dismantled and put back into operation in the same as well as in different environments. Further, we investigate the model performance on different pumps from the same type compared to those from the training data.**

***Index Terms*—Internet of Things (IoT), Industry and Production 4.0, Predictive Maintenance**

I. Introduction

The NAMUR open architecture (NOA) enables the monitoring and optimization sensors of existing "brownfield" plants in the process industry. It sketches a second data channel in addition to existing core process control systems like Simatic PCS neo. With the help of low cost multi-sensors, previously non instrumented assets can be retrofitted with a communication layer. This enables monitoring and classification of the operational states and anomalies of an asset based on the retrofitted sensor measurements. Machine learning models have great potential for these classification tasks for each individual asset in a production plant. The performance of machine learning models, however, strongly depends on the available training data and respective data distribution. Acquiring training data and labels through measurements of normal and anomaly conditions of industrial assets is expensive and very time-consuming. Anomaly conditions may negatively impact these monitored assets by down wearing or wrecking. Further, it is often difficult to transfer the models from one asset to another one, even if those are from the same type and highly standardized. Numerous real-world factors, as minimal divergence in production, wear and tear, or environmental factors influences the data distribution recorded by a sensor and consequently the derived predictions of machine learning models. Further, the data distribution is influenced by the sensor itself with several degrees of freedom in position and rotation. On the other hand, the same, industry-standard assets are often used in production facilities around the world. For example, having hundreds of the same type of filling pumps in one food and beverage plant is very common. Therefore, a robust and reusable model for a specific asset type is required for an application of economic condition monitoring. In other words, a single robust machine learning model, which can be delivered together with a respective sensor.

Therefore, aim of this paper are the following: (i) introduce the IIoT measurement system and deployment, (ii) analyze challenges deriving requirements for a robustness and scalable model dissemination, (iii) systematically evaluate the impact and challenges of production divergence, wear and tear, and maintenance in the context of robustness and transferability in machine learning models (iv) compare a feature-based Neural Network approach and ROCKET, including different pre-processing and post-processing combinations (v) derive initial guidelines to address these complex classification problem.

II. Related Work

A. IIoT Sensor Systems

In the era of industry 4.0, Industrial IoT (IIoT) sensor devices are increasingly used to monitor and adapt to changes in the environment [1]–[3]. Sensors can capture a variety of physical values as light, temperature, pressure, vibration, and sound. The information is linked to the IIoT network to share data and connect between devices and management systems. The correct operation of IIoT sensors play an essential role in the overall system performance [4]. Several IIoT sensor kits exist [1], [2], but only a few are appropriate for industrial conditions. In the industrial context, IIoT sensor devices are often deployed in harsh environments, with ambient temperature, humidity, and strong vibrational conditions [5]. Further, a simple communication and a long battery and overall sensor lifespan is required [5]. Additionally, sensors within the IIoT should be high quality and low cost so that they can be used in very large numbers and enable the data collection from the variety of physical values simultaneously. The SITRANS multi sensor was specifically developed for industrial applications and its requirements [6]. Further, the compromise between high sampling rate and long sampling duration required for an accurate model prediction and limited data acquisition required for a long battery lifespan still remains an open challenge [3].

B. Time Series Classification

Several traditional and deep learning based approaches for time series classification have been explored over the last

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_2

decade [7]–[10]. Often methods are tested on the UCR/UEA archive, an open collection of over 150 different data sets, having almost 20 sensor data sets available, with currently no data set in the automation or IIoT context. Deep learning approaches such as Convolutional Neural Networks (CNNs) and deep Residual Networks are less computational expensive and showed to be the most promising in a recent systematic method study on the UCR/UEA data set [7], [9], [10]. In a recent paper [8], ROCKET a simple linear classifier based on random convolutional kernel transformations showed a comparable high accuracy to Neural Networks (NN) at the UCR/UEA data set, but required only fraction of the computational expense compared to existing methods [8]. While, in general, these methods show great potential for time series classification, they have only been developed and tested within public repositories as the UCR/UEA archive [11]. Besides the impressive number and amount of data collected, the transfer and robustness towards similar assets and the influence of maintenance work or environmental conditions have not been considered yet. However, it is important to evaluate whether a model dissemination from a lab to a real-world scenario can be achieved with the method developed.

III. IIoT Measurement System

A. Industrial Asset

An industrial asset can be any kind of machine or mechanical device that uses power to apply forces and control movement to perform an intended action. This asset can vary in size, purpose, and placement condition. Typical examples are pumps, motors, or manufacturing robots. An asset can be placed in a lot of scenarios: a motor pump combination can be placed outdoors or be part of a large bottling plant.

B. Sensor Measurement System

With the help of low-cost multi-sensors measurement system, previously non instrumented assets can be retrofitted with a dedicated communication channel (Fig 1). These sensors can record airborne, structure-borne sound or temperature data, for example. We suggest that condition and the future behavior of any assets can be assessed using system-specific or central functions and methods (advanced analytic, scheduling). The respective experiments in this context have taken place only with the structure-bone data. These data offer a sampling rate of 6644 Hz. In order to keep the sensor battery lifespan as long as possible, the data is only collected for 512 samples every min, i.e. a total of 77 ms. Variations in the signal are observed due to any variations in the asset, the environment conditions of the asset, the asset health status and due to the sensor mounting and rotation. The sensor uses a Bluetooth low energy (BLE) wireless interface to communicate with a gateway device. Data collection is performed periodically for a short period of time. Within a fraction of a sec, the 512 samples are collected. Afterwards, the sensor goes into sleep mode, thereby saving energy. The gateway is connected to the internet and transmits the data via the Message Queuing Telemetry Transport (MQTT) protocol to the cloud-based backend services. Both, the number of samples and the data collection interval can be adjusted depending on the monitoring requirements. These requirements depend on the actual operational behavior of the asset.

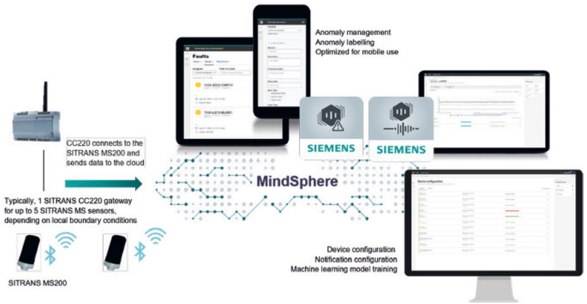

Anomaly management
Anomaly labelling
Optimized for mobile use
Faults
SIEMENS
SIEMENS
CC220 connects to the SITRANS MS200 and sends data to the cloud
MindSphere
Typically, 1 SITRANS CC220 gateway for up to 5 SITRANS MS sensors, depending on local boundary conditions
SITRANS MS200
Device configuration
Notification configuration
Machine learning model training

Fig. 1. IIoT measurement system and deployment

C. Machine Learning System and Deployment

1) Training: The raw sensor data is stored in a blob-store in the cloud. The user can use a web dashboard to load the sensor data for a particular asset and sensor. The web dashboard is used by the machine operator to visually analyze and label the data. The dashboard offers a wizard-like approach to select the labelled data set and use a machine learning template to trigger the training. A template is a predefined machine learning algorithm and a set of hyperparameters that can be fine-tuned. The template abstracts the details of the algorithm so that non-ML experts can use the system. The actual training is done on a high-performance compute cluster. The cluster offers state-of-the-art elastic scaling capabilities, which is an essential requirement for large-scale sensor deployments. The final model is saved in the blob store. Model metrics are saved in a model lifecycle management database and can be visualized in the dashboard.

2) Prediction: The user can view the metrics and perform the model deployment. The system loads the model from the blob store and provisions the model to a runtime service called inference server. The inference server handles the execution of multiple concurrent models by subscribing to live sensor data in the cloud and passing the data to corresponding models. Pre- and post-processing steps are described at a later point in the paper. Prediction results are handled by a rule-based system. For example, a detected anomaly is routed to the notification application. The user loads an anomaly documentation application to view and validate anomaly detection or classification results.

D. Model Dissemination and Requirements

Migrating a data science model from a research lab to a real-world deployment is non-trivial and potentially a continuous, ongoing process. Consequently, many machine learning models never go into production. A major challenge in industrial setups is the positive economic impact of a machine learning model. The economic impact can be considered positive when the costs and effort to create, deploy and update the model are below the savings gained by the model. Therefore, it is important to have a scalable machine learning solution that can be used on the same asset types in various environments. A scalable model can be considered as a digital product where replications are easy to achieve and can broadly distributed to the same asset-sensor combination anywhere in the world. This would result in high costs for the initial phase and marginal costs for each additional model application, mainly driven by the deployment infrastructure and low-cost sensors.

A machine learning model can be considered as scalable when replications are easy to achieve and identical copies can broadly be distributed to the same asset-sensor combination anywhere in the world. This can be achieved in the two main machine learning paradigms, supervised and unsupervised learning, in different ways. In an unsupervised machine learning approach, the collection of training data is mostly automated, does not require any labels and is consequently time and cost wise inexpensive. However, unsupervised learning can only be used in specific anomaly detection tasks. In the supervised learning approach, the training data requires labels, which are very hard to collect, especially in classes reflecting anomaly behavior. Therefore, in a supervised learning approach, scalability is achieved by an increased robustness of the machine learning model, so it can be transferred easily to any other asset. Our aim is to thrive forward towards an all-in-one solution machine learning model solution applicable for one asset type sensor combination. Therefore, we suggest employing the advantages of both, supervised and unsupervised learning through an ensemble based model voting of an anomaly classification and an anomaly detection model.

Further, we believe that a stable prediction may be more important compared to an instant prediction. This reasoning is derived by the assumption, that in real-world applications, a normal or anomaly class of an industrial asset will be given through a longer period. Therefore, a smoothing filter for the resulting model probabilities is suggested. Another important point is to lower the entry barrier as far as possible for the user. The user, ideally the domain expert, using the machine learning models for holistic asset monitoring, should be enabled to carry out all sensor and model related tasks using the web dashboard (Fig 1). The installation of the sensor should also be as simple as possible. An exact sensor position can be specified on an asset, while an exact sensor rotation is challenging. Consequently, a virtual sensor alignment is required.

IV. Proposed Machine Learning Pipeline

The proposed machine learning pipeline consists of 1) virtual sensor alignment 2) employment of a general classification model learning from all classes 3) post-processing of the classification output 4) specific model learning from healthy class and 5) the model voting (Fig 2). Virtual sensor alignment ensures the rotational invariance of the sensor by remapping the sensor from the sensor coordinate system which can differ between sensors into a unified virtual one. The generic classification model is trained with all classes and is assumed to be trained long-term on an increased number of data from various different assets. An autoencoder can be employed as specific detection model and only trained on healthy data. As this data is relatively inexpensive to collect, the autoencoder is assumed to be trained individually for each asset. Two different approaches were explored, a) a feature-based ANN and b) an end-to-end approach ROCKET. From the resulting Logits the probabilities of both approaches were derived and smoothed with a moving average filter. Finally, voting between generic classification and specific detection model was applied.

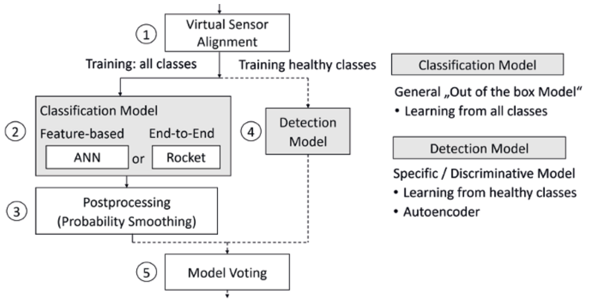


Fig. 2. Proposed machine learning pipeline

1) Virtual Sensor Alignment. To increased robustness and transferability, rotational invariance of the machine learning model input data is suggested. We propose remapping the data from sensor coordinate system s_i into the world coordinate system w_i using the Kabsch algorithm [12] minimizing the following loss function L which is solved for the rotation matrix C (to align s_i to w_i):

$$L(C) = \sum_{i=1}^{n} ||S_i - C_{w_i}||^2 \tag{1}$$

2) Classification Model.

a) Feature-based Approach: ANN. The ANN approach employed a sliding window on the time series data of the virtual axes, followed by the extraction of the Mel-frequency cepstral coefficients (mfccs), minority oversampling and normalization (Fig 3). The resulting 20 coefficients were feed into the ANN.

Data Augmentation and Feature Selection. Besides more widely used as features for audio classification, we suggest mfccs for vibration data as both having strong relation due to air-borne and structure-borne sound transmission [13]. The mfccs represent the power spectrum of the short-term Fourier transform on a nonlinear frequency scale inspired by human biology, uniformly spaced below 1 kHz and logarithmic scale above 1 kHz. Two different data augmentation techniques are suggested, a) sliding window on time series data and b) minority oversampling on the extracted features. Sliding window: The 1x512 input values are artificially increased towards 16x256 with window size of 256 and on offset of 16. Minority oversampling: Collecting data from anomaly classes is challenging and often has a detrimental effect on an industrial asset. Therefore, synthetic minority oversampling technique (SMOTE) [14] of the mfccs features is proposed in the pipeline.

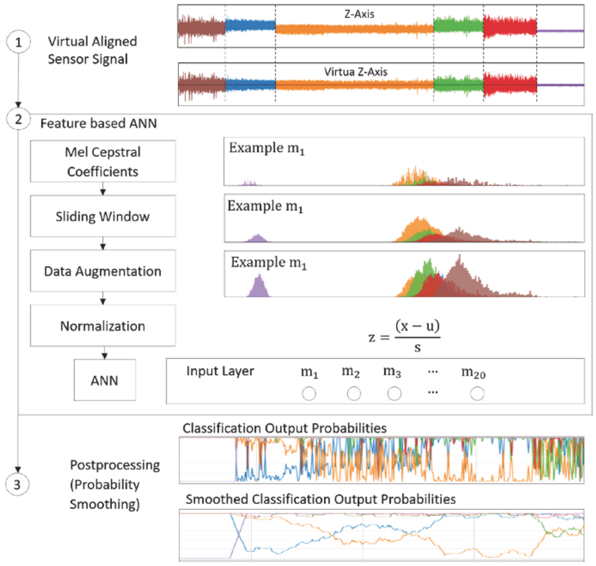


Fig. 3. Data visualization for 1) virtual Sensor alignment, 2) feature-based ANN and 3) probability smoothing of the proposed machine learning pipeline

Hyperparameter Consideration. When designing the architecture for an ANN, a variety of parameters can be tuned. The art is to find the right combination for these parameters to achieve the highest accuracy and lowest loss. Therefore, TALOS [15] was employed for the hyperparameter search. The framework allows to randomly sample from a given grid of hyperparameters and train the respective networks to support the user finding the best combination. The following parameters resulted from the framework: The respective α: 0.001, β: 0.9 β_1: 0.9, β_2:0.999, ϵ: 10^{-8} for Adam optimization, # layers: 2, # hidden units: 64 (each hidden layer), relu activation function, mini batch size: 16 and a SoftMax output layer. The TALOS framework, however, does not allow to design best network parameters for training and test data from different distribution. Consequently, this resulted in a very small bias, but high variance as the model was overfitting to the training data. Therefore, we employed dropout of 40% and early stopping after 30 epochs. The resulting mccfs were normalized between [-1,1] based on a transformation resulting in a Gaussian distribution before fed into the Neural Network.

A. End-to-End-Approach: ROCKET.

The end-to-end machine learning model ROCKET [8] transforms the virtual aligned sensor signal using a large number of convolutional kernels. The convolutional kernels are randomly created varying length, weights, bias, dilation, and padding. The transformed features were used to train a linear ridge classifier. The classifier relies on cross-validation and L2 regularization to avoid overfitting to the training data, accepting a defined bias to reduce the variance. The combination of ROCKET and regression forms, in effect, a single-layer convolutional neural network with random kernel weights, where the transformed features form the input for a trained SoftMax layer [8].

3) Output Probability Considerations. As the network prediction only takes the current point in time into account, which have been observed to fluctuate quite strongly (Fig 3, Classification Output Probabilities, Fig 4, (2)), we employed a filter to smooth the probabilities. In this case a moving average filter [size: 15] was employed (Fig 3, Smoothed Classification Output Probabilities, Fig 4, (3)). Since it is a SoftMax layer, raw probabilities can be acquired similar to the ANN and ROCKET approach.

4) Detection Model. It can be assumed that an unlimited amount of healthy training data is available for an industrial asset. This data can be used for asset specific anomaly detection to increase the overall robustness of our ML pipeline. Therefore, we trained an unsupervised machine learning model to detect the anomalies in our dataset by using an autoencoder architecture based fully connected deep neural network (DNN). The autoencoder learns to reconstruct the input for healthy sensor data, as it was trained to do so, but will fail to reconstruct anomaly data. The reconstruction error, the error between input and output signal, was used as an anomaly score. The threshold was calculated as mean + standard deviation of the reconstruction error of healthy data. The DNN architecture consists of five fully connected layers with the respective number of neurons per layers #512 input #256 encoding 126 bottleneck #256 decoding #512 output; $tanh$ was employed as activation function.

5) Model Voting. The model voting was done in a way, that the autoencoder always overruled the classification results and a non-healthy class had to change to the anomaly class with the highest probability and vice versa, if the autoencoder predicted the other class (Fig 4, (5)).

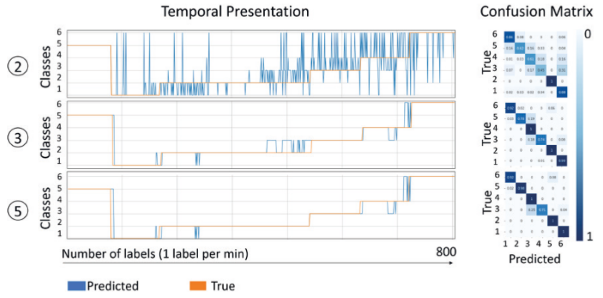


Fig. 4. Prediction results at different steps within the proposed machine learning pipeline. Training data 1; example On/Off, feature-based ANN approach

V. Measurement Setup and Data Set Description

All measurements were done in a laboratory test bench. As an industrial asset, a centrifugal pump motor combination was selected. Two series of measurements with each one training data set and four test data sets were created. The first series of measurements was created to examine the robustness of a model during maintenance work or sensor battery change within the same pump, and the second to examine how well the models can be transferred to other pumps of the same construction.

Measurement Series I (within pump):

- Training Set I: Reference
- Test Set 1 Asset was turned off and cooled down
- Test Set 2: Sensor was removed and reattached
- Test Set 3: Screws were removed and reattached
- Test Set 4: Asset was completely dismantled and rebuilt

Measurement Series II (between pumps):

- Training Set II: Pump X
- Test Set 5: Pump I
- Test Set 6: Pump II
- Test Set 7: Pump III
- Test Set 8: Pump IV

The data sets consisted of six classes, i.e., three healthy operational conditions and three non-healthy operational conditions.

Healthy (Normal) Operational Conditions:

- Class 1, normal load (flow rate of $50\,\mathrm{m}^3\,\mathrm{h}^{-1}$)
- Class 2, partial load (flow rate of $12.5-\ 37.5\,\mathrm{m}^3\,\mathrm{h}^{-1}$)
- Class 6, idle state (flow rate of $0\,\mathrm{m}^3\,\mathrm{h}^{-1}$)

Non-Healthy (Anomaly) Conditions:

- Class 3, dry running pump (flow rate of $0\,\mathrm{m}^3\,\mathrm{h}^{-1}$)
- Class 4, hydraulic blockage (flow rate of $0\,\mathrm{m}^3\,\mathrm{h}^{-1}$)
- Class 5, cavitation (nominal point 50 $60\,\mathrm{m}^3\,\mathrm{h}^{-1}$, net positive suction head (NPSH) of -0.8)

Each class was measured at least 30 min per pump, corresponding to 30 labels. The measurements were acquired on different days and even months, but according to the same measurement protocol. As anomalies were introduced manually, there are deviations within the classes. This, however, can be considered as a real-world problem, as those data are also expected to be broader distributed.

VI. Machine Learning Experiments

We aimed to evaluate the impact and challenges of production divergence and maintenance in the context of robustness and transferability systematically. Therefore, we performed an initial baseline experiment where training and test set were derived from the same data sets for Training Data I and II. Subsequently, we performed systematic experiments: first, we trained one Model I on Training Set I and Model II on Training Set I and II. The trained models were tested on all eight test sets. To compare different pre-processing algorithms and machine learning models, we evaluated all experiments with an ANN and ROCKET. Both methods were evaluated without virtual sensor alignment (M), with virtual sensor alignment (V+M), with subsequent probability smoothing (V+M+S) and subsequent employing of the autoencoder (V+M+S+A). The ideal criteria for using an autoencoder, that training and test set come from the same pump where no maintenance work has been performed on either the pump or the sensor, was met only by test set 1.

Hypothesis were tested with paired or unpaired t-test (depending on the comparison).

VII. Experimental Results and Conclusion

The baseline classification accuracy was quite high (>0.972) for both data sets (table 1). ROCKET showed slightly higher performance and was essential faster during the training (but not prediction).

TABLE I
Accuracy, Baseline Experiments

	ANN		ROCKET	
	M	V+M	M	V+M
Training Data I	0.976	0.978	0.983	0.973
Training Data II	0.972	0.973	0.986	0.986

M-model; V-virtual sensor alignment

Results for the robustness and transferability experiments are shown in table 2.

1) Feature-based ANN vs ROCKET approach: The average accuracy over all robustness and transferability experiments was higher in the ROCKET approach (0.862±0.116) compared to the feature-based ANN (0.857±0.078), the difference was statistically significant (p=0.002, paired t-test) (Model Voting was excluded as available for one experiment only). Including only results with respective pre- and post-processing steps (virtual sensor alignment and probability smoothing) the differences vanished: ROCKET approach (0.909±0.104) compared to ANN (0.908±0.063) (p=0.965, paired t-test).

2) Same vs Different Measurement Series: The impact of using the same measurement series vs a different one is clearly high, with an average accuracy of both models 0.978±0.005 on the baseline experiments vs 0.829±0.067 on the transferability and robustness experiments without pre- and post-processing (unpaired t-test: <0.001). Pre- and post-processing, however, decreased the difference for average results of both models (0.908±0.068), but remained significant (unpaired t-test: <0.015).

3) Same vs Different Pump Series: Average accuracy classification accuracy was notably higher in same pumps (0.848±0.062) vs different pumps (0.809±0.069) but did not reach statistically significance (p=0.055, unpaired t-test). The difference was somewhat lower (but still notable even though significance vanished even more), employing pre- and post-processing methods: same pumps (0.924±0.058) vs different pumps (0.892±0.105), (p=0.154, unpaired t-test).

4) Sensor Alignment: Without post-processing, the sensor alignment had a slightly negative impact with averaged model results without (0.829±0.069) vs with (0.811±0.096) virtual sensor alignment (p=0.05); with post-processing the relationship turned around model results without (0.889±0.106) vs with (0.908±0.086) virtual sensor alignment (p=0.06). Note: Sensor position was attached according to the same protocol and sensor was not rotated intentionally.

TABLE II
ACCURACY, ROBUSTNESS AND TRANSFERABILITY EXPERIMENTS

Feature-based Approach: ANN								
	Within Pumps				Between Pumps			
	T1	T2	T3	T4	T5	T6	T7	T8
Model I (Training Data I)								
M	0.842	0.804	0.884	0.788	0.809	0.755	0.816	0.825
MS	0.917	0.929	0.951	**0.800**	**0.819**	0.822	**0.992**	0.905
V**M**	0.852	0.822	0.888	0.616	0.822	0.783	0.843	0.807
V**MS**	0.928	**0.931**	**0.974**	0.788	**0.819**	**0.953**	**0.992**	**0.921**
V**M**SA	**0.985**							
Model II (Training Data I + Training Data II)								
M	0.861	0.801	0.876	0.721	0.811	0.772	0.825	0.922
MS	0.928	**0.904**	**0.938**	0.804	0.819	0.819	**0.990**	**0.975**
V**M**	0.849	0.787	0.849	0.637	**0.821**	0.818	0.796	0.884
V**MS**	0.913	0.891	0.912	**0.813**	0.819	**0.982**	0.912	0.972
V**M**SA	**0.980**							
End-End-based Approach: ROCKET								
Model I (Training Data I)								
M	0.893	0.850	0.934	0.760	0.779	0.778	0.661	0.725
MS	0.920	**0.936**	0.994	0.646	0.795	0.923	0.523	**0.806**
V**M**	0.882	0.846	0.944	0.835	**0.822**	0.779	0.518	0.658
V**MS**	0.917	0.934	**0.995**	**0.939**	0.819	**0.951**	**0.622**	0.742
V**M**SA	**0.994**							
Model II (Training Data I + Training Data II)								
M	0.910	0.895	0.952	0.801	0.782	0.909	0.840	0.933
MS	0.991	0.975	0.993	**0.904**	0.803	0.984	**0.987**	0.966
V**M**	0.909	0.887	**0.944**	0.637	**0.821**	0.887	0.848	0.876
V**MS**	0.990	**0.986**	0.992	0.884	0.819	**0.998**	**0.987**	**0.967**
V**M**SA	**0.998**							

Rem-Removal and Installation; T-Test; M-Model; V-Virtual Sensor Alignment S-Smoothing of Probabilities; A-Autoencoder; Training data 1: smaller model; Training data 2: larger, more diverse model

5) Probability Smoothing: The impact of probability smoothing is clearly high on all averaged model accuracy before smoothing (0.812±0.063) and after smoothing (0.901±0.066) and highly significant ($p<0.001$, paired-t-test).

6) Variation in Training Data: There was no worsening nor any improvement when the training data was either only acquired from the same pump (training data 1: 0.873±0.087) or from the same and a different pump (training data 1: 0.879±0.091) ($p<0.308$, paired-t-test). However, when training and test measurements were acquired from different pumps, the more complex training data had significantly positive impact on the accuracy: average accuracy training data 1 (0.800±0.087) vs training data 2 (0.879±0.091) ($p<0.001$, paired-t-test).

7) Impact of Model Voting: The model voting employing the anomaly detection results of an autoencoder, trained on normal data only, increased the classification accuracy in all cases explored (test set 1 only): without (0.937±0.036) with (0.989±0.008) ($p<0.042$, paired-t-test).

VIII. CONCLUSION AND FUTURE WORK

This work is limited to a single asset type and a single environment. The influence of these factors may be investigated in future studies. A limitation of the study is the small numbers of models compared. However, the aim was to provide some initial guideline for the overall machine learning pipeline.

First, we endorse a fixed sensor position. The virtual sensor alignment showed a small impact, however, in all measurements, the sensor position was attached according to the same protocol and not rotated intentionally. Such an exact placement of the sensor may not always be the case in a real-world scenario. Therefore, the virtual alignment is recommended. Additionally, we suggest the employment of probability smoothing as post-processing. The approach can be used independent of the model, as the raw probabilities can be acquired from the Logits in each model using a SoftMax layer for classification. Additionally, we recommend increasing the complexity of the training data. Further, a specific autoencoder, trained on easy-to-collect healthy data combined with the classification results is proposed.

REFERENCES

[1] X. Tong, H. Yang, L. Wang, and Y. Miao, "The Development and Field Evaluation of an IoT System of Low-Power Vibration for Bridge Health Monitoring," *Sensors*, vol. 19, no. 5, p. 1222, mar 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/5/1222

[2] X. Zhao, G. Wei, X. Li, Y. Qin, D. Xu, W. Tang, H. Yin, X. Wei, and L. Jia, "Self-powered triboelectric nano vibration accelerometer based wireless sensor system for railway state health monitoring," *Nano Energy*, vol. 34, pp. 549–555, apr 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2211285517301143

[3] T. Schneider, N. Helwig, S. Klein, and A. Schütze, "Influence of sensoring rate on multivariate statistical condition monitoring of industrial machines and processes," *Proceedings*, vol. 2, no. 13, 2018. [Online]. Available: https://www.mdpi.com/2504-3900/2/13/781

[4] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, jul 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0007681315000373

[5] V. C. Gungor and G. P. Hancke, "Industrial wireless sensor networks: Challenges, design principles, and technical approaches," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 4258–4265, 2009.

[6] T. Bierweiler, H. Grieb, S. von Dosky, and M. Hartl, "Smart Sensing Environment – Use Cases and System for Plant Specific Monitoring and Optimization," *Automation*, pp. 155–158, 2019. [Online]. Available: https://elibrary.vdi-verlag.de/index.php?doi=10.51202/9783181023518-155

[7] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, "Deep learning for time series classification: a review," *Data Mining and Knowledge Discovery*, vol. 33, no. 4, pp. 917–963, 2019.

[8] A. Dempster, F. Petitjean, and G. I. Webb, "ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels," *Data Mining and Knowledge Discovery*, vol. 34, no. 5, pp. 1454–1495, sep 2020. [Online]. Available: http://link.springer.com/10.1007/s10618-020-00701-z

[9] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, may 2017, pp. 1578–1585. [Online]. Available: http://ieeexplore.ieee.org/document/7966039/

[10] F. Karim, S. Majumdar, H. Darabi, and S. Chen, "LSTM Fully Convolutional Networks for Time Series Classification," *IEEE Access*, vol. 6, pp. 1662–1669, 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8141873/

[11] A. Bagnall, J. Lines, W. Vickers, and E. Keogh, "The UEA & UCR Time Series Classification Repository." [Online]. Available: www.timeseriesclassification.com

[12] F. L. Markley, "Attitude Determination Using Vector Observation: A Fast Optimal Matrix Algorithm," *The Journal of the Astronautical Sciences*, vol. 41, no. 2, pp. 261–280, 1993.

[13] L. Cremer, M. Heckl, Petersson, and B. A.T., *Structure-Borne Sound - Structural Vibrations and Sound Radiation at Audio Frequencies*. Springer, 2005.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, jun 2002. [Online]. Available: https://www.jair.org/index.php/jair/article/view/10302

[15] Autonomio Talos [Computer software]. (2019). Retrieved from http://github.com/autonomio/talos.

Data-driven Cut-off Frequency Optimization for Biomechanical Sensor Data Pre-Processing

Severin Bernhart*, Verena Venek*, Christina Kranzinger*, Wolfgang Kremser* and Aaron Martínez†
*Salzburg Research FGmbH, 5020 Salzburg, Austria
†University of Salzburg, 5400 Hallein, Austria
{severin.bernhart, verena.venek, christina.kranzinger, wolfgang.kremser}@salzburgresearch.at,
aaron.martinez@sbg.ac.at

***Abstract*—The pre-processing of biomechanical sensor data often involves signal filters for noise removal in order to improve the performance of segmentation and machine learning algorithms. However, finding an optimal value for the filter's cut-off frequency is time consuming, as researchers have to rely on heuristics and experience. Therefore, we introduce a method called F_cOpt for automatically estimating an optimal cut-off frequency for noise filtering in one-dimensional biomechanical data. The method resamples the input data and applies three automated cut-off frequency determination methods, pools their individually suggested cut-off frequencies with a k-means cluster algorithm and provides an optimal cut-off frequency for filtering one-dimensional data streams. We demonstrate F_cOpt in the context of a ski turn segmentation algorithm. This methodology counteracts the susceptibility for incongruously identifying cut-off frequencies by automated methods caused by high sampling rates. F_cOpt suggests a cut-off frequency of 2.63 Hz instead of the originally proposed 3 Hz. Filtering with the suggested cut-off frequency on average deviates from the original temporal accuracy of the ski turn segmentation by 1.0 ms, which corresponds to only 0.08% in relation to the mean turn duration. Although F_cOpt cannot entirely replace heuristics for cut-off frequency determination yet, it is an easy tool for researchers who want to improve the signal pre-processing for their segmentation algorithms. It lays the groundwork for future developments in the area of data-driven filter design.**

***Index Terms*—cut-off frequency optimization, automated filtering techniques, clustering algorithm, human motion data analytics**

I. Introduction

Knowledge extraction from biomechanical sensor data involves a number of steps, the first of them being pre-processing [1], [2]. The main purpose is de-noising, i.e. the removal of unwanted artefacts from the signal. These artefacts are caused by electromagnetic or -static interferences on the sensor data [3]. For successive processing steps, cleansed training and test data improves the accuracy of segmentation algorithms and machine learning models [4]. The most common method to filter unwanted frequencies in time series is to apply digital filtering algorithms. Their application requires the definition of the filter type, filter order and in particular the cut-off frequency F_c, which defines the threshold for removal. Finding an optimal F_c is a time-consuming pre-processing step in data analysis and requires experience and domain knowledge. The cut-off frequency has to be adjusted to the certain use case, including the subsequent segmentation or classification task and, thus, is hardly generalizable [5]. So far, in the field of human motion data analysis, the process of differentiating noise from the signal has mostly been a heuristic approach performed by experts. However, automatic methods have been developed for supporting this heuristic and time-consuming F_c determination process, in particular for the initial exploratory data analysis. Within the last decades, for biomechanical data the residual analysis (RA) of Winter [6] and Wells [7] and the power spectral analysis (PSA) of D'Amico M. and Ferrigno [8] were repeatedly compared in literature [5], [9]–[11]. The comparisons investigated the performance of the methods on kinematic walking and running data from lower extremities. Giakas and Baltzopoulos [9] identified the PSA as best method in their work. For the evaluation, they used the root mean squared error between the filtered signal with their optimal F_c and the reference signal. Sinclair, Taylor and Hobbs [10] did not yield a preference between the PSA and the RA. They claimed that the PSA requires a predefined threshold, but suggested using 95% of the signal power for the F_c determination based on their outcomes. Aissaoui, Husse, Mecheri, Parent and de Guise [12] compared PSA and an autocorrelation (AC) function based on knee rotation data. They stated that the AC function of Challis [13] behaves well compared to spline functions. The described methods aim to find the F_c that filters out a high amount of noise without removing any power of the signal. Nevertheless, the authors could not generally commit to one method as the optimal automated filtering technique. Giakas and Baltzopoulos mentioned that an appropriate application of automated filtering methods require expert knowledge about the limitation of automated methods and signal characteristics [9]. In addition, Mullineaux [11] and Fazlali, Sadeghi, Saba, Ojaghi and Allard [5] stated that the performance of automated methods is highly dependent on the type of the input data and underlined that higher sampling rates of the input signals result in a higher cut-off frequency outcome. The Shannon theorem [14] states that a signal must be sampled with at least twice of its signal frequency in order to be reproducible. Additionally, the authors Fazlali, Sadeghi, Saba, Ojaghi and Allard [5] and Campbell, Bradshaw, Ball, Hunter and Spratford [15] remarked that F_c is dependent on individual properties of humans and the motion primitive (i.e. the motion segment), which is aimed to be determined. Recent

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_3

literature of Inertial Measurement Unit (IMU) data analysis in skiing exemplifies the discrepancies in filter selection procedures. Whereas Martínez et al. [16] and Danielsen, Sandbakk, McGhie and Ettema [17] presented examples where F_c was determined heuristically, Klous, Müller and Schwameder [18] and Reid, Haugen, Gilgien, Kipp and Smith [19] used the residual AC method of Challis [13] to identify the optimal cut-off frequency. Meland [20] performed Welch's power spectral density analysis to identify a proper F_c, which is similar to the power spectral analysis of D'Amico M. and Ferrigno [8] but additionally contains a heuristic determination of F_c. Therefore, this paper presents a method called F_cOpt to help finding an optimal F_c for signal-noise separation by clustering the results of three commonly used and automated cut-off frequency determination methods in biomechanical data pre-processing with a k-means cluster algorithm. We validated the method on a gyroscope signal coming from an IMU attached onto ski boots and compared the outcome to the heuristic cut-off frequency determination process originally reported in Martínez et al. [16].

II. Methods

A. The F_cOpt method

F_cOpt consist of two components: F_c-aggregator and F_cOpt-identifier. In order to determine the optimal cut-off frequency F_c for one-dimensional time series, three automated methods were applied to the raw sensor signal, which were further pooled with a cluster algorithm to determine an optimal F_c (see Fig. 1):

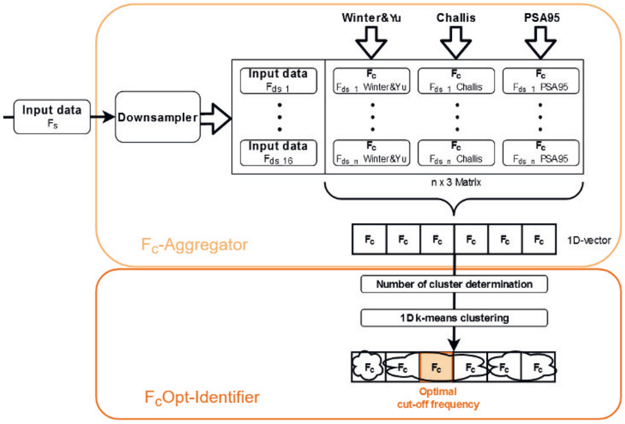


Fig. 1. F_cOpt method overview separated into two components: F_c-Aggregator and F_cOpt-Identifier

1) `Winter&Yu`: Residual analysis of Winter [6] and Wells [7] with the regression extension of Yu, Gabriel, Noble and An [21]: A predefined set of possible cut-off frequencies is determined in the range of zero and half of the sampling rate F_s. The input signal is filtered with a zero-lag second order Butterworth filter for each F_c, respectively. The residuals are computed between the input signal and each filtered signal. A regression line is calculated with the residuals computed at the cut-off frequencies $\frac{F_s}{10}$ and $\frac{F_s}{2} - 5$ [21]. The intersection of the regression line and the residual y-axis determines consequently the optimal F_c.
2) `Challis`: Autocorrelation (AC) function by Challis [13]: The filtered signals for a predefined set of possible cut-off frequencies are computed as described in (1). The AC function is defined as the average product of a signal and a shifted version of itself. The AC function is applied several times on a predefined set of lag values. The AC function outputs are summed up for each filtered signal. The F_c that corresponds to the minimal summed up value is determined as the optimal F_c.
3) `PSA95`: Power spectrum analysis of D'Amico M. and Ferrigno [8]: A Fast Fourier Transform (FFT) is performed on the input signal. The optimal F_c is determined by selecting the frequency in which 95% of the signal power is contained. The percentage was set to 95% according to the suggestion of Sinclair, Taylor and Hobbs [10].

As the methods of Winter&Yu and Challis require further preparation to control the settle phases, a zero-padding of 20% of the input data length was added at the beginning and end of each input signal [22] before the automated cut-off frequency determination methods were applied in the F_c-aggregator. As mentioned in the introduction, all methods depend on the sampling rate (F_s) of the input time series: higher sampling rates cause a larger amount of noise in the data. This results in an increased F_c output of the exemplified automated filtering methods. The generalized F_cOpt method solved this issue by applying the cut-off frequency determination methods on downsampled input time series. Each downsampled sampling rate (F_{ds}) was in the interval $I_{ds} = [1; F_s]$. Setting the lower limit of I_{ds} to 1 Hz enabled the detection of very low F_c–values (close to 0 Hz) because the input signal was downsampled to a F_{ds} close to 1 Hz. This definition intercepted if the initial F_s had been set exceedingly higher than the recommendation by the Shannon theorem [14]. Defining the upper limit of I_{ds} to F_s enabled the detection of an optimal F_c close to $\frac{F_s}{2}$ (see definition in (1)), which caught the case that the initial F_s had been approximately set to Shannon's suggestion. Therefore, an optimal F_c could be determined in the range of 0 Hz and $\frac{F_s}{2}$. Moreover, exceeding the upper limit of I_{ds} to higher sampling rates would have required an oversampling of the initial signal and added noise that would have unnecessarily increased the F_cOpt output. In the range of the interval I_{ds}, 16 equally separated F_{ds} were determined based on pilot tests on the dataset. Subsequently, the interval I_c was computed for each F_{ds}, respectively, because Winter&Yu and Challis additionally need this predefined set of possible F_c. For each of the 16 F_{ds}, a cut-off frequency was determined out of I_c, which was divided into hundred evenly distributed cut-off frequencies. Thus, a matrix containing the F_c–values was created with the shape of 16 × 3 derived from the number of investigated sampling rates and the three determination methods. Firstly, the automated methods Winter&Yu, Challis and PSA95 were executed on the dataset with the median F_{ds} of the interval I_{ds}. Then, the median F_{ds} was discarded from I_{ds} and the methods were applied again to the next median F_{ds} of I_{ds}. The algorithm converges when $F_{ds} < \frac{50}{4}$ holds. This threshold is derived from Winter&Yu's method, as it ensures that $\frac{F_s}{10} < \frac{F_s}{2} - 5$ and $f_{\mathrm{c,b}} < f_{\mathrm{c,e}}$ hold [21]. This convergence criterion prevents the formation of clusters in a very low range of $\mathrm{F_c}$ close to 0 Hz. Hence, the shape of the matrix was $n \times 3$ with $n \leq 16 \wedge n \in N$. In order to determine the optimal $\mathrm{F_c}$ of the signal, the resultant matrix was converted to a one-dimensional vector. All matrix values were inserted into a one-dimensional vector sorted in ascending order, with the $\mathrm{F_c}$ losing their information about their automated method. In the $\mathrm{F_c}$Opt-identifier, a cluster algorithm was applied to detect groups of similar $\mathrm{F_c}$ in the vector. In advance, the number of clusters had to be specified. Therefore, an adopted version of the k-means cluster algorithm by Wang and Song [23] was used, which is applicable to one-dimensional data and inherits the number of cluster determination. It defines the number of clusters using the Bayesian information criterion to subsequently separate the $\mathrm{F_c}$ in the defined number of clusters. This method was chosen because it was described in the documentation [23] as an optimal, fast and reproducible univariate clustering method. The centers of the clusters signified possible $\mathrm{F_c}$Opt outcomes. Finally, the $\mathrm{F_c}$Opt method returned the center of the largest cluster as optimal $\mathrm{F_c}$. Fig. 2 shows the $\mathrm{F_c}$-aggregation after applying the automated methods to the downsampled time series and the subsequent $\mathrm{F_c}$Opt-identification in the center of the largest cluster of the one-dimensional vector.

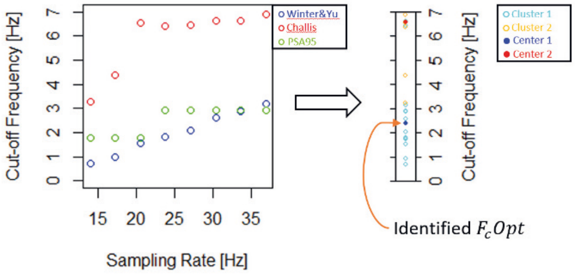


Fig. 2. Example of applying $\mathrm{F_c}$Opt on one dataset.

B. Evaluation

The $\mathrm{F_c}$Opt method was validated regarding two aspects:

1) Robustness against data sampling rate to evaluate if $\mathrm{F_c}$Opt was able to accommodate input signals with high sampling rates;
2) The accuracy of the $\mathrm{F_c}$Opt determination compared to a heuristic cut-off frequency determination process for the application of segmentation algorithms.

The kinematic human motion data of Martínez et al. [16] was used for the evaluation. The authors gathered three-dimensional accelerometer and gyroscope data from an IMU attached onto each (left and right) ski boot while one participant performed "ski turns" on a ski-ergometer. The resulting dataset consisted of twelve data signals resampled to 50 Hz. The aim of their work was to validate a ski turn segmentation methodology. Furthermore, the authors heuristically investigated the optimal $\mathrm{F_c}$ for their segmentation algorithm development. They compared the temporal accuracy of the segmentation algorithm when it was applied to the available IMU signals, as well as, to the averaged data of the left and right boot for each signal. The dataset contained 124 skiing turns. Different conditions regarding speed and slope were simulated to add robustness to the developed algorithm: two different time durations based on the average duration of giant slalom and slalom turns (1.45 and 0.90 sec, respectively), and three different inclinations of the ski-ergometer [16]. Thus, the whole dataset consists of six runs. The authors gained the highest accuracy by applying the segmentation algorithm to the averaged gyroscope's z-axis (GYRO-Z), i.e. the roll axis of the sensor, when the data stream was low-pass filtered with a fourth-order zero-lag Butterworth filter with a $\mathrm{F_c}$ of 3 Hz [16]. Hence, the GYRO-Z signal was used in the current work to validate the $\mathrm{F_c}$Opt method.

1) Sampling rate robustness: The GYRO-Z data of the presented dataset were resampled to sampling rates in the range of 15 and 50 Hz stepped by 5 Hz, in the range of 50 and 400 Hz stepped by 25 Hz and in the range of 400 and 1000 Hz stepped by 100 Hz. Linear interpolation was used for upsampling the dataset above 50 Hz, whereby the results rely on estimation. Since 12.5 Hz fulfills the condition (1) required for the application of Winter&Yu, the minimal starting sampling rate of 15 Hz was chosen. The selected ranges differ in order to investigate the behaviour of the methods for lower and very high sampling rates. The 100 Hz steps at the higher frequencies were selected due to computation efficiency. After resampling, the $\mathrm{F_c}$Opt, Winter&Yu, Challis and PSA95 were applied to the dataset, respectively. For each automated method, optimal $\mathrm{F_c}$-values were determined over the different sampling rates. The second derivative of the generated $\mathrm{F_c}$-signals was computed to identify the critical sampling rate. The first peak in the second derivative implies a strong increase of the optimal $\mathrm{F_c}$, i.e. characteristic for the loss of a stable state or rather the critical sampling rate for each method.

2) Temporal accuracy evaluation: The segmentation algorithm by Martínez et al. [16] was validated comparing the timestamps detected by the algorithm to the expert-labelled skiing turn switches based on video data. The absolute time difference between the detected turn switches and the limits of agreement (LoA) of the expert-labelled timestamps was computed. This time difference was presentative for the temporal accuracy, which was decisive for the determination of an optimal $\mathrm{F_c}$. Between the five evaluated $\mathrm{F_c}$ ([0, 3, 6, 9, 12] Hz), 3 Hz was defined as an optimal cut-off frequency for the segmentation task. In the current study, F_cOpt was applied to the same dataset with a sampling rate of 50 Hz to match the resampling in the original experiment. The 3 Hz of the low-pass filter was replaced by the F_cOpt's output and the temporal accuracy of the segmentation algorithm was compared with the results of Martínez et al. [16]. The temporal difference was computed in average per turn. In addition, the absolute values were presented in relation to the mean turn duration. In order to evaluate the performance of F_cOpt, statistical metrics, such as the standard deviation of the temporal differences, were computed. A comparison was rendered to evaluate if the segmentation algorithm was more biased or contained outliers after changing the F_c, e.g. turn switches were detected too early or too late.

III. Results

A. F_cOpt sampling rate robustness

Table I and Fig. 3 present the optimal F_c outputs of the automated methods dependent on the sampling rate of the input signals. All methods resulted in a rising optimal F_c when the sampling rate increased. Winter&Yu, Challis and PSA95 varied between [0.85 – 93.44] Hz, [3.52 – 183.50] Hz and [1.73 – 37.31] Hz, respectively. The F_cOpt output provided values between 0.00 Hz and 31.79 Hz. Fig. 5 underlines that F_cOpt produced an output very close to 0 Hz until a F_s of 25 Hz. Whereas, PSA95 and Challis showed an irregularly increasing slope, the Winter&Yu method delivered a linear increasing F_c with rising F_s. Furthermore, Fig. 4 presents that the median optimal F_c of Winter&Yu, Challis and PSA95 were 17.83 Hz, 52.28 Hz and 10.02 Hz, respectively. F_cOpt provided a median value of 5.98 Hz. The methods also differed in their deviation of the lower and upper quartiles of the F_c output. Whereas Winter&Yu, Challis and PSA95 yielded an interquartile range of 32.91 Hz, 105.20 Hz and 25.56 Hz, respectively, F_cOpt provided a deviation of 3.85 Hz. Fig. 6 illustrates the second derivative of the methods' outputs, showing large peaks at 125 Hz and 250 Hz for Challis. PSA95 displays a single peak at 200 Hz, as well as F_cOpt and Winter&Yu at 400 Hz.

B. F_cOpt temporal accuracy evaluation

For the turn segmentation algorithm of Martínez et al. [16], the F_cOpt proposed an optimal cut-off frequency of 2.63 Hz for the sampling rate of 50 Hz. In comparison to the given accuracy of 6.6 ms in average per turn (mean bias = 0.2 ms, LoA = 56.6 ms) [16], the application of the F_cOpt-determined F_c resulted in an absolute difference of 5.6 ms (mean bias = -0.2 ms, LoA = 53.7 ms). This absolute difference corresponded to 0.48% of the mean turn duration of 1.175 sec compared to the 0.56% of Martínez et al. [16]. F_cOpt deviates only by 1.0 ms per turn (0.08% of the mean turn duration) in the temporal accuracy compared to the result of Martínez et al. [16].

The comparison between the suggested F_c of Martinez et al. [16] and F_cOpt yielded a difference of 0.37 Hz. The standard deviation of the time differences per turn was 18.7 ms for the F_cOpt's outcome in comparison to the estimated F_c of 19.0 ms [16]. This resulted in an increased standard deviation by 0.3 ms with the application of F_cOpt. Moreover, the maximum temporal deviation for a specific turn was 80 ms in both cases.

TABLE I
F_cOPT OUTCOMES FOR THE DEFINED SAMPLING RATES (F_S) (ALL IN HZ)

	Methods			
F_s	*Winter&Yu*	*Challis*	*PSA95*	*F_cOpt*
15	0.85	3.52	1.73	0.00
20	1.47	5.17	2.20	0.00
25	1.98	5.45	2.39	0.00
30	2.53	6.21	2.55	0.89
35	3.07	6.34	2.59	2.55
40	3.54	6.94	2.70	1.84
45	4.06	7.39	2.87	2.43
50	4.51	7.07	3.02	2.63
75	5.69	7.83	3.39	3.01
100	6.49	7.92	4.22	4.59
125	8.11	14.31	4.70	4.79
150	10.23	48.74	5.46	5.08
175	12.67	49.94	6.03	5.09
200	16.33	51.52	6.66	6.05
225	19.32	53.03	13.36	6.08
250	22.73	53.87	24.05	5.94
275	25.47	98.84	26.47	6.01
300	28.79	108.08	27.36	6.10
325	31.19	109.15	27.98	6.22
350	33.59	110.48	28.31	6.41
375	35.99	112.06	28.51	6.33
400	38.39	112.79	28.51	6.47
500	47.98	125.84	29.35	16.33
600	57.07	152.53	31.19	21.94
700	59.52	166.75	34.72	29.15
800	65.32	172.39	35.93	26.67
900	80.31	178.03	36.98	31.41
1000	93.44	183.50	37.31	31.79

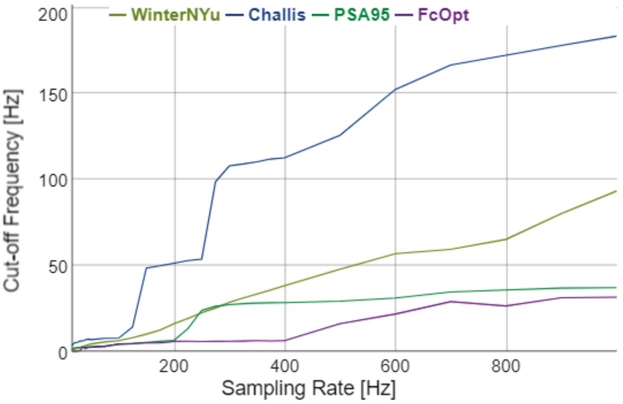


Fig. 3. Comparison of the sampling rate robustness in the range of from 15 to 1000 Hz.

IV. Discussion

The results of the sampling rate robustness experiment illustrated that the F_c interquartile range of the F_cOpt outputs was the smallest compared to the other assessed automated methods for the sampling rates between 15 Hz and 1000 Hz. The Challis method was found to be mostly dependent on the

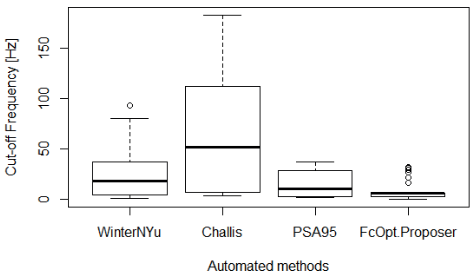


Fig. 4. Distributions of the $\mathrm{F_c}$-results compared between the automated methods in the sampling rate range between 15 Hz and 1000 Hz.

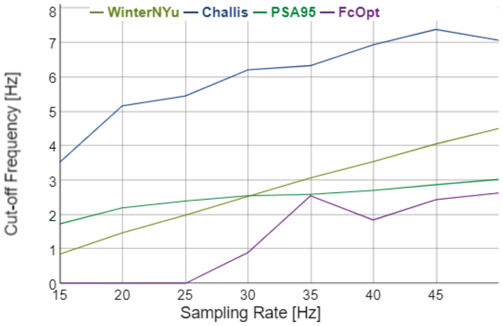


Fig. 5. Comparison of the sampling rate robustness in a closer look in the range from 15 and 50 Hz.

sampling rate, while $\mathrm{F_c}$Opt was the most robust one against data sampling rates. The results confirmed that $\mathrm{F_c}$Opt can be applied on datasets with high sampling rates in contrast to the other exemplified automated methods. Applying Winter&Yu, Challis and PSA95 on differently sampled data increased the probability of finding a properly sampled input data setting for each method. Therefore, the methods' outputs coincided at a certain frequency range. Selecting the center of this frequency range as optimal $\mathrm{F_c}$ is more reliable than choosing a $\mathrm{F_c}$ based on one method's output on a predefined sampling rate. As the first peak is decisive for the loss of the stable state, only Winter&Yu showed a peak later at 400 Hz in Fig. 6. However, since the $\mathrm{F_c}$ of Winter&Yu in Fig. 3 and 5 increased continuously, the loss of the stable state of this method cannot be clearly detected on this experiment. In summary, these aspects highlight the improved robustness of $\mathrm{F_c}$Opt against high sampling rates.

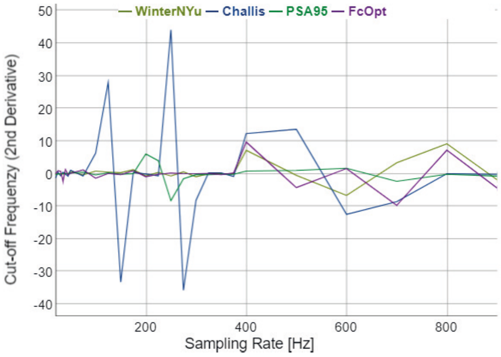


Fig. 6. Comparison of the second derivative of the automated methods' outputs.

The optimal $\mathrm{F_c}$ determined by $\mathrm{F_c}$Opt only slightly deviated from the segmentation algorithm outcome of Martínez et al. [16]. Thus, $\mathrm{F_c}$Opt is a $\mathrm{F_c}$ proposing method, but does not fully replace the heuristic determination process. Nevertheless, $\mathrm{F_c}$Opt definitely can reduce the time spent by researchers to evaluate possible cut-off frequencies, in particular, when familiarizing with sensor signals. Therefore, the proposed approach can be integrated into data processing and preparation platforms, e.g. the extended human motion activity recognition chain [1].

The influence of the size of the interval $\mathrm{I_{ds}}$ ($\mathrm{F_{ds}}$-resolution) and the pre-defined vector of possible $\mathrm{F_c}$ ($\mathrm{F_c}$-resolution) required for Winter&Yu and Challis on the $\mathrm{F_c}$Opt-outcome was additionally investigated. Table II and Table III present that the output differed between 0.38 Hz for the $\mathrm{F_{ds}}$-resolution and 0.31 Hz for the $\mathrm{F_c}$-resolution. While the $\mathrm{F_c}$ did not converge to a certain value for the $\mathrm{F_c}$-resolution comparison, the $\mathrm{F_c}$ converged at approximately 2.5 Hz for a $\mathrm{F_{ds}}$-resolution of 32 steps and higher. Hence, $\mathrm{F_c}$Opt is only slightly affected by altering the $\mathrm{F_{ds}}$-resolution and $\mathrm{F_c}$-resolution.

$\mathrm{F_c}$Opt does not deliver above 0 Hz until a sampling rate of 25 Hz is reached (see Fig. 5). This can be prevented by starting applying Winter&Yu, Challis and PSA95 on the data with the maximum F_{ds} of the interval I_{ds}. Therefore, $\mathrm{F_c}$Opt provides results for data that at least meet the threshold $F_s < \frac{50}{4}$ required for $\mathrm{F_c}$Opt to apply.

The proposed method was validated on gyroscope data of an IMU sensor attached onto ski boots for alpine skiing turns. It must be investigated if $\mathrm{F_c}$Opt is applicable for other kinematic sensor data and movement primitives despite skiing turns. The methods of Winter&Yu, Challis and PSA95 are established methods in the field of kinematic data. Nevertheless, the integration of recent automated $\mathrm{F_c}$ determination methods into the $\mathrm{F_c}$Opt approach from other research areas than data science in sport would be beneficial and should be investigated. The other way around, the application of the $\mathrm{F_c}$Opt method in other research areas could benefit the initial estimation and determination of a cut-off frequency to start processing and mining available sensor signals.

TABLE II
$\mathrm{F_C}$OPT OUTCOME DEPENDENCY ON $\mathrm{F_{DS}}$-RESOLUTION.

$\mathrm{F_{ds}}$-resolution	$\mathrm{F_c}$Opt output
4	2.76 Hz
8	2.38 Hz
16	2.63 Hz
32	2.55 Hz
64	2.54 Hz

TABLE III
F$_{\mathrm{C}}$OPT OUTCOME DEPENDENCY ON F$_{\mathrm{C}}$-RESOLUTION.

F$_{\mathrm{c}}$-resolution	**F$_{\mathrm{c}}$Opt output**
25	2.47 Hz
50	2.76 Hz
100	2.63 Hz
200	2.57 Hz
400	2.45 Hz

V. CONCLUSION

This work developed and validated a generalized method that can be used as a data-driven cut-off frequency determination method for signal and noise separation when sampling rates above 25 Hz are given. The validation dataset illustrated that F$_{\mathrm{c}}$Opt can approach the heuristic F$_{\mathrm{c}}$-determination process but not completely replace it. Nevertheless, F$_{\mathrm{c}}$Opt can save time resources by supporting experts with data preprocessing. Additionally, the exemplified method helps non-experts for the determination of a proper cut-off frequency for rapid noise filtering and data smoothing. Particularly, only the sampling rate of the dataset must be known to apply the method. Furthermore, F$_{\mathrm{c}}$Opt has potential to be integrated in real-time applications to implement segmentation algorithms that require an adaptive filter for non-stationary signal processing.

ACKNOWLEDGMENT

This work was partly funded by the Austrian Federal Ministry for Transport, Innovation and Technology, the Austrian Federal Ministry for Digital and Economic Affairs, and the federal state of Salzburg.

REFERENCES

[1] R. Brunauer, W. Kremser, and T. Stöggl, "From sensor data to coaching in alpine skiing–a software design to facilitate immediate feedback in sports," in *International Symposium on Computer Science in Sport*. Springer, 2019, pp. 86–95.

[2] A. Holzinger, "Introduction to machine learning & knowledge extraction (make)." *Machine learning and knowledge extraction*, vol. 1, no. 1, pp. 1–20, 2019.

[3] S. Khemiri, K. Aloui, and M. S. Naceur, "Preprocessing of biomedical signals: Removing of the baseline artifacts," *2013 10th International Multi-Conference on Systems, Signals and Devices, SSD 2013*, pp. 3–7, 2013.

[4] J. A. Sáez, J. Luengo, and F. Herrera, "Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification," *Pattern Recognition*, vol. 46, no. 1, pp. 355–364, 2013.

[5] H. Fazlali, H. Sadeghi, S. Saba, M. Ojaghi, and P. Allard, "Comparison of four methods for determining the cut-off frequency of accelerometer signals in able-bodied individuals and acl ruptured subjects," *Gait & Posture*, 2020.

[6] D. Winter, "Assessment of signal and noise in the kinematics of normal, pathological and sporting gaits," in *Proceedings of the First Biannual Conference of the Canadian Society of Biomechanics*, vol. 1, no. 8, 1975, pp. 307–320.

[7] R. P. Wells, "Assessment of signal and noise in the kinematics of normal, pathological and sporting gaits," *Human locomotion*, pp. 92–93, 1980.

[8] M. D'amico and G. Ferrigno, "Technique for the evaluation of derivatives from noisy biomechanical displacement data using a model-based bandwidth-selection procedure," *Medical and Biological Engineering and Computing*, vol. 28, no. 5, pp. 407–415, 1990.

[9] G. Giakas and V. Baltzopoulos, "A comparison of automatic filtering techniques applied to biomechanical walking data," *Journal of Biomechanics*, vol. 30, no. 8, pp. 847–850, 1997.

[10] J. Sinclair, P. J. Taylor, and S. J. Hobbs, "Digital filtering of three-dimensional lower extremity kinematics: An assessment," *Journal of human kinetics*, vol. 39, no. 1, pp. 25–36, 2013.

[11] D. R. Mullineaux, "Using a breakpoint to determine the optimal cut-off frequency," *ISBS Proceedings Archive*, vol. 35, no. 1, p. 140, 2017.

[12] R. Aissaoui, S. Husse, H. Mecheri, G. Parent, and J. A. de Guise, "Automatic filtering techniques for three-dimensional kinematics data using 3d motion capture system," in *2006 IEEE International Symposium on Industrial Electronics*, vol. 1. IEEE, 2006, pp. 614–619.

[13] J. H. Challis, "A procedure for the automatic determination of filter cutoff frequency for the processing of biomechanical data," *Journal of Applied Biomechanics*, vol. 15, no. 3, pp. 303–317, 1999.

[14] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, 1949.

[15] R. A. Campbell, E. J. Bradshaw, N. Ball, A. Hunter, and W. Spratford, "Effects of digital filtering on peak acceleration and force measurements for artistic gymnastics skills," *Journal of Sports Sciences*, pp. 1–10, 2020.

[16] A. Martínez, R. Jahnel, M. Buchecker, C. Snyder, R. Brunauer, and T. Stöggl, "Development of an automatic alpine skiing turn detection algorithm based on a simple sensor setup," *Sensors*, vol. 19, no. 4, p. 902, 2019.

[17] J. Danielsen, Ø. Sandbakk, D. McGhie, and G. Ettema, "Mechanical energetics and dynamics of uphill double-poling on roller-skis at different incline-speed combinations," *PloS one*, vol. 14, no. 2, p. e0212500, 2019.

[18] M. Klous, E. Müller, and H. Schwameder, "Collecting kinematic data on a ski/snowboard track with panning, tilting, and zooming cameras: is there sufficient accuracy for a biomechanical analysis?" *Journal of sports sciences*, vol. 28, no. 12, pp. 1345–1353, 2010.

[19] R. C. Reid, P. Haugen, M. Gilgien, R. W. Kipp, and G. A. Smith, "Alpine ski motion characteristics in slalom," *Frontiers in Sports and Active Living*, vol. 2, p. 25, 2020.

[20] H. J. Meland, "Automated detection and classification of movement cycles in cross-country skiing through analysis of inertial sensor data movement patterns," Master's thesis, NTNU, 2017.

[21] B. Yu, D. Gabriel, L. Noble, and K.-N. An, "Estimate of the optimum cutoff frequency for the butterworth low-pass digital filter," *Journal of Applied Biomechanics*, vol. 15, no. 3, pp. 318–329, 1999.

[22] S. J. Howarth and J. P. Callaghan, "The rule of 1 s for padding kinematic data prior to digital filtering: Influence of sampling and filter cutoff frequencies," *Journal of Electromyography and Kinesiology*, vol. 19, no. 5, pp. 875–881, 2009.

[23] H. Wang and M. Song, "Ckmeans. 1d. dp: optimal k-means clustering in one dimension by dynamic programming," *The R journal*, vol. 3, no. 2, p. 29, 2011.

[24] Lim, Kim, and Park, "Prediction of Lower Limb Kinetics and Kinematics during Walking by a Single IMU on the Lower Back Using Machine Learning," *Sensors*, vol. 20, no. 1, p. 130, dec 2019. [Online]. Available: https://www.mdpi.com/1424-8220/20/1/130

[25] T. S. Yoo, S. K. Hong, H. M. Yoon, and S. Park, "Gain-scheduled complementary filter design for a MEMS based attitude and heading reference system," *Sensors*, vol. 11, no. 4, pp. 3816–3830, 2011.

[26] G. Christodoulakis, K. Busawon, N. Caplan, and S. Stewart, "On the filtering and smoothing of biomechanical data," *2010 7th International Symposium on Communication Systems, Networks and Digital Signal Processing, CSNDSP 2010*, pp. 512–516, 2010.

[27] E. W. Forgy, "Cluster analysis of multivariate data: efficiency versus interpretability of classifications," *biometrics*, vol. 21, pp. 768–769, 1965.

[28] A. Cappello, P. F. La Palombara, and A. Leardini, "Optimization and smoothing techniques in movement analysis," *International Journal of Bio-Medical Computing*, vol. 41, no. 3, pp. 137–151, 1996.

[29] F. Alonso, J. Del Castillo, and P. Pintado, "Application of singular spectrum analysis to the smoothing of raw kinematic signals," *Journal of biomechanics*, vol. 38, no. 5, pp. 1085–1092, 2005.

Deep Learning based Anomaly Detection and Scene Classification

A Low-Complexity Deep Learning Framework For Acoustic Scene Classification

Lam Pham*, Hieu Tang†, Anahid Jalali*, Alexander Schindler*, Ross King* and Ian McLoughlin‡
*Austrian Institute of Technology, 1210 Vienna, Austria
†Hongik University, 04066 Seoul, Korea
‡Singapore Institute of Technology, 138683, Singapore, Singapore
{lam.pham, seyedehanahid.naghibzadehjalali, alexander.schindler, ross.king}@ait.ac.at,
tqhieu94@gmail.com, ian.mcloughlin@singaporetech.edu.sg

***Abstract*—In this paper, we presents a low-complexity deep learning frameworks for acoustic scene classification (ASC). The proposed framework can be separated into three main steps: Front-end spectrogram extraction, back-end classification, and late fusion of predicted probabilities. First, we use Mel filter, Gammatone filter, and Constant Q Transform (CQT) to transform raw audio signals into spectrograms, where both frequency and temporal features are presented. Three spectrograms are then fed into three individual back-end convolutional neural networks (CNNs), classifying into ten urban scenes. Finally, a late fusion of three predicted probabilities obtained from three CNNs is conducted to achieve the final classification result. To reduce the complexity of our proposed CNN network, we apply two compression techniques: model restriction and decomposed convolution. Our extensive experiments, which are conducted on DCASE 2021 (IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events) Task 1A Development and Evaluation datasets, achieve a low-complexity CNN based framework with 128 KB trainable parameters and the best classification accuracy of 66.7% and 69.6%, improving DCASE baseline by 19.0% and 24.0% respectively.**

***Index Terms*—Convolutional neural network, Gammatone filter, constant Q transform, MEL filter, spectrogram, deep learning models.**

I. Introduction

Acoustic Scene Classification (ASC), one of main research fields of ‘Machine Hearing’ [1], has drawn much attention in recent years and has been applied to a wide range of real-life applications such as enhancing the listening experience of users by detecting scene context [2], [3], supporting sound event detection [4], or integrating in a multiple-sensor automatic system [5]. To deal with ASC challenges such as unbalanced data, lacking data input, or mismatched recording devices, various methods have been proposed, which can be separated into two main approaches. The first approach makes use of multiple data input such as ensemble of spectrograms [6]–[9] or audio channels [10]. Meanwhile, the second approach focuses on back-end classification, and proposes powerful deep learning network architectures, which are able to enforce the training process [11]–[15]. Although these two approaches can achieve good results, they present high-complexity systems. Indeed, while multiple input data requires an ensemble of multiple individual classification models [16], [17], powerful network architectures show a number of convolutional layers [13], [14]. All top-10 systems proposed in recent DCASE challenges in 2018, 2019, 2020 are also based on complex architectures, requiring larger than 2 MB of trainable parameters. The issue of large model prevents implementing edge devices concerning real-life applications which require a low footprint. Although there are various methods proposed to deal with the issue of model complexity such as quantization [18], pruning [19], model restriction (i.e. restriction on the number of layers [20], the number of kernel [21], or both of these factors [22]), decomposed convolution [23], hybrid methods using pruning and decomposed convolution [23], pruning and distillation [24], these are mainly applied for image data. Therefore, our work introduces a low-complexity deep learning framework for ASC. To deal with ASC challenges, we propose an ensemble of multiple spectrogram inputs, using Mel filter [25], Gammatone filter [26], and CQT [25]. Regarding each network used for training an individual spectrogram input, we deal with the issue of model complexity by combining model restriction and decomposed convolution methods.

The remaining of our paper is organized as follows: Section 2 presents proposed deep learning frameworks and model compression techniques. Section 3 introduces evaluation setup where experimental setting, metric, and implementation of deep learning frameworks proposed are presented. Next, Section 4 presents and analyses the experimental results. Finally, Section 5 presents conclusion and future work.

II. The low-complexity deep learning framework proposed

A. *Our baseline*

We first propose a baseline with a high-level architecture shown in Fig. 1, which comprises three main steps in the order of front-end spectrogram feature extraction, back-end classification, and a fusion of predicted probabilities. At the first step, a raw audio signal is firstly transformed into a spectrogram of 128 × 704 by using 128 MEL filter [25] with Fast Fourier Transform (FFT) number, window size, and hope size set to 8192, 4096, and 620, respectively. As delta and delta-delta across the temporal dimension are applied to the spectrogram, we then generate the spetrograms of

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_4

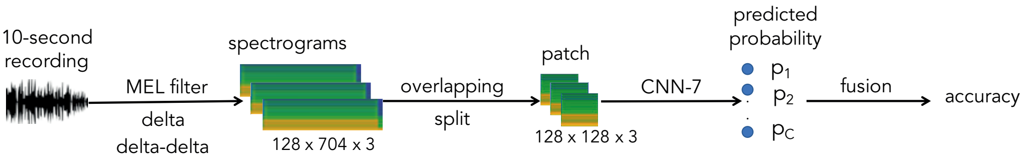


Fig. 1. High-level architecture of ASC baseline system proposed.

TABLE I
THE CNN-7 NETWORK ARCHITECTURE BASELINE (INPUT PATCH OF 128×128×3)

Network architecture	**Output**
BN - Convolution ([3×3]@$C_{out}1 = 32$) - ReLU - BN - Dropout (10%)	128×128×32
BN - Convolution ([3×3]@$C_{out}2 = 32$) - ReLU - BN - AP [2×2] - Dropout (10%)	64×64×32
BN - Convolution ([3×3]@$C_{out}3 = 64$) - ReLU - BN - Dropout (10%)	64×64×64
BN - Convolution ([3×3]@$C_{out}4 = 64$) - ReLU - BN - AP [2×2] - Dropout (10%)	32×32×64
BN - Convolution ([3×3]@$C_{out}5 = 128$) - ReLU - BN - AP [2×2] - Dropout (10%)	16×16×128
BN - Convolution ([3×3]@$C_{out}6 = 128$) - ReLU - BN - GAP - Dropout (10%)	128
FC - Softmax	$C = 10$

$128 \times 704 \times 3$. Next, the spectrograms are split into 10 patches of $128 \times 128 \times 3$ with 50% overlapping before feeding into a CNN based network for classification.

As we illustrate our proposed CNN based network architecture in Table I, it contains sub-blocks, which perform convolution with $C_{out}K$ channel (Convolution ([kernel size]@$C_{out}K$, $1 \leq K \leq 6$), batch normalization (BN) [27], rectified linear units (ReLU) [28], average pooling (AV [kernel size]), global average pooling (GAP), Dropout (percentage dropped) [29], fully-connected (FC), and Softmax layers. The dimension of Softmax layer is set to $C = 10$ that equals to the number of scene context classified. In total, we have 6 convolutional layers and 1 fully-connected layers that makes the proposed network architecture like CNN-7.

As the CNN-7 works on patches, the final predicted probability of an entire spectrogram is computed by averaging of all patches. Let us consider $\mathbf{P^n} = (\mathbf{p_1^n}, \mathbf{p_2^n}, ..., \mathbf{p_C^n})$, with C being the category number and the n^{th} out of N patches fed into the CNN-7, as the probability of all patches, then the average classification probability is denoted as $\mathbf{\bar{p}} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_C)$ where,

$$\bar{p_c} = \frac{1}{N}\sum_{n=1}^{N} p_c^n \quad for \quad 1 \leq n \leq N \tag{1}$$

and the predicted label $\hat{y}$ of the entire spectrogram is determined as:

$$\hat{y} = argmax(\bar{p}_1, \bar{p}_2, ..., \bar{p}_C) \tag{2}$$

B. Ensemble of multiple spectrogram inputs

Although both of CQT and STFT spectrograms are built on Fourier Transform theory, they extract different central frequencies. Meanwhile, both Mel spectrogram and Gammatonegram are generated from STFT spectrogram, but use two different filter banks: Mel and Gammatone filters. We can conclude that three spectrograms either extract different central frequencies or apply different auditory models. Therefore, each spectrogram may contain its own distinct and complimentary information. This inspires us to propose an ensemble of these three spectrograms as a rule of thumb to improve the ASC performance [16], [17]. To evaluate the ensemble of multiple spectrograms, we proposed a late fusion scheme, referred to as PROD fusion. In particular, we conduct experiments over individual networks with different spectrogram inputs, then obtain predicted probability of each network as $\mathbf{\bar{p}_s} = (\bar{p}_{s1}, \bar{p}_{s2}, ..., \bar{p}_{sC})$, where C is the category number and the s^{th} out of S networks evaluated. Next, the predicted probability after PROD fusion $\mathbf{p_{f-prod}} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_C)$ is obtained by:

$$\bar{p_c} = \frac{1}{S}\prod_{s=1}^{S} \bar{p}_{sc} \quad for \quad 1 \leq s \leq S \tag{3}$$

Finally, the predicted label $\hat{y}$ is determined by (2).

C. Model compression methods applied to the CNN-7 network

Our proposed single CNN-7 architecture reports a complexity of 1,129 MB for non-zero parameters with using 32 bits for representing one parameter. Additionally, using ensemble of three spectrogram inputs makes the number of trainable parameters further increase three times. To reduce the model complexity, we firstly restrict the number of channels used in the CNN-7 baseline, then reduce the channels of $C_{out}1$ from 32 to 16, $C_{out}3$ and $C_{out}4$ from 64 to 32, $C_{out}5$ and $C_{out}6$ from 128 to 64. Our proposed channel restriction (CR) helps to reduce an individual CNN-7 complexity to 313 KB that nearly equals to 1/4 of the original size.

We further reduce the CNN-7 complexity by applying the decomposed convolution (DC) technique described in [23], [30]. Let us consider C_{in} and C_{cout} as the input and output channel numbers respectively, $W = 3$ and $L = 3$ are the dimensions of kernel size, which are used for a convolutional layer. Then, the number of trainable parameters at a convolutional layer is computed by $W \times L \times C_{in} \times C_{out} = 9 \times C_{in} \times C_{out}$. We reduce the number of trainable parameters at a convolutional layer by decomposing the convolutional

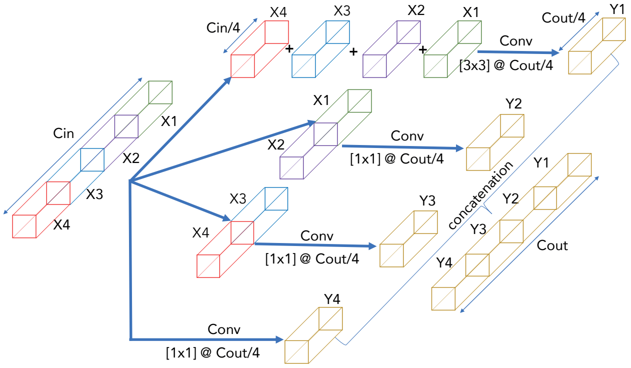


Fig. 2. Decomposed convolution technique applied to a convolutional layer.

TABLE II
THE NUMBER OF 10-SECOND AUDIO RECORDINGS CORRESPONDING TO EACH SCENE CATEGORIES IN THE TRAIN. AND EVAL. SUBSETS SEPARATED FROM THE DCASE 2021 TASK 1A DEVELOPMENT DATASET [35], AND THE EVALUATION DATASET [32].

Category	**Train. Subset**	**Eval. Subset**	**Evaluation**
Airport	1393	296	-
Bus	1400	297	-
Metro	1382	297	-
Metro Station	1380	297	-
Park	1429	297	-
Public square	1427	297	-
Shopping mall	1373	297	-
Street pedestrian	1386	297	-
Street traffic	1413	297	-
Tram	1379	296	-
Total	13962 (≈38.79 hours)	2968 (≈8.25 hours)	7920 (22 hours)

layer into four sub-convolutional layers as described in Fig. 2. For all four sub-convolutional layers, the output channel is reduced to $C_{out}/4$. Regarding the first sub-convolutional layer (the upper path shown in Fig. 2), although we still use kernel size of [W×L]=[3×3], we reduce the input channels to C_{in}/4, then cost $(9 \times C_{in} \times C_{out})/16$ trainable parameters. Regarding the other sub-convolution layers, we reduce the kernel size to [W×L]=[1×1]. While the second and third sub-convolutional layers (two middle paths shown in Fig. 2), the input channel is reduced to $C_{in}/2$, it is remained in the fourth sub-convolutional layer (the lower path shown in Fig. 2). As a result, it requires $(C_{in} \times C_{out})/8$ for the second and third sub-convolutional layers, and $(C_{in} \times C_{out})/4$ for the fourth sub-convolution layer. By decomposing a convolutional layer into four sub-convolutional layers, the model complexity is reduced to nearly 1/8.5 of the original size. By combining the two model compression techniques, we can achieve a CNN-7 network architecture with complexity of 42.6 KB, which nearly equals to 1/25 of the original size (i.e. the original CNN-7 network architecture is proposed in the baseline framework in Table I). As we need to use three CNN-7 for three different spectrogram inputs, the final complexity of the proposed framework is approximately 128 KB.

III. EVALUATION SETTING

A. TAU Urban Acoustic Scenes 2020 Mobile, the Development [31] and Evaluation [32] datasets (DCASE 2021 Task 1A)

The **Development dataset** was proposed for DCASE 2021 Task 1A challenge [33], which requires a limitation of model complexity set to 128 KB with using 32 bits for one trainable parameter. The dataset in slightly unbalanced, recorded from 12 large European cities: Amsterdam, Barcelona, Helsinki, Lisbon, London, Lyon, Madrid, Milan, Prague, Paris, Stockholm, and Vienna. It consists of 10 scene classes: airport, shopping mall (indoor), metro station (underground), pedestrian street, public square, street (traffic), traveling by tram, bus and metro (underground), and urban park. The audio recordings were recorded from 3 different devices namely A (10215 recordings), B (749 recordings), C (748 recordings). Additionally, synthetic data for 11 mobile devices was created based on the original recordings, referred to as S1 (750 recordings), S2 (750 recordings), S3 (750 recordings), S4 (750 recordings), S5 (750 recordings), and S6 (750 recordings).

As a result, this task not only requires a low complexity model, but it also proposes an issue of mismatch recording devices. To evaluate, we follow the DCASE 2021 Task 1A challenge [33], separate this dataset into training (Train.) and evaluation (Eval.) subsets as shown in Table II. Then, Train. subset is used for training the framework proposed and Eval. subset is used for evaluating. Notably, two of 12 cities and S4, S5, S6 audio recordings are only presented in the Eval. subset for evaluating the issue of mismatch recording devices and unseen samples.

Furthermore, the DCASE 2021 Task 1A challenge releases the **Evaluation dataset** without labels, which is used to evaluate the submitted systems. The total number of 10-s segments is 7920 (22 hours), which is significantly larger than the Development dataset. In this paper, our results on both Eva. subset and Evaluation dataset (i.e. Accuracy scoring on Evaluation datset is conducted by DCASE 2021 task 1A challenge as labels is not published) are reported and compared with the state-of-the-art systems.

B. Deep learning framework implementation

We use Tensorflow framework to build all classification models in this papers. The cross-entropy loss function used for training is described as

$$LOSS_{EN}(\Theta) = -\frac{1}{N}\sum_{n=1}^{N}\mathbf{y_n}\log\left\{\hat{\mathbf{y}}_\mathbf{n}(\Theta)\right\} + \frac{\lambda}{2}||\Theta||_2^2 \quad (4)$$

defined over all parameters Θ, and N is the number of training samples. λ denotes the ℓ_2-norm regularization coefficient. $\mathbf{y_n}$ and $\hat{\mathbf{y}}_\mathbf{n}$ denote ground truth and predicted output. The training is carried out for 100 epochs using Adam [34] for optimization.

TABLE III
Performance comparison of CNN-7 w/ CR & DC among three spectrograms, with different time lengths, with or without using data augmentations (Acc. %)

	Without data augmentations				**With Data augmentations**			
Spectrogram	**1-second**	**2-second**	**5-second**	**10-second**	**1-second**	**2-second**	**5-second**	**10-second**
MEL	56.7	57.9	56.2	60.5	54.6	57.9	59.5	58.4
GAM	53.2	55.0	53.1	53.9	58.9	60.1	60.6	57.1
CQT	44.3	47.7	48.6	49.2	44.2	45.7	48.6	49.1

TABLE IV
Performance comparison among DCASE baseline, the CNN-7 baseline, the CNN-7 baseline with channel restriction (CNN-7 w/ CR), the CNN-7 baseline with channel restriction and decomposed convolution (CNN-7 w/ CR & DC) (Acc. %).

Category	**DCASE baseline (90.3 KB)**	**CNN-7 baseline (1.1 MB)**	**CNN-7 w/ CR (313 KB)**	**CNN-7 w/ CR & DC (42.6 KB)**
Airport	40.5	59.5	50.3	64.5
Bus	47.1	73.7	70.4	69.0
Metro	51.9	57.6	49.8	70.0
Metro station	28.3	53.9	48.1	45.1
Park	69.0	73.1	78.5	74.4
Public square	25.3	34.3	38.4	25.9
Shopping mall	61.3	52.9	50.2	43.4
Street pedestrian	38.7	39.4	35.0	32.7
Street traffic	62.0	84.5	88.2	89.6
Tram	53.0	67.9	62.5	52.7
Average	47.7	59.7	57.1	56.7

C. Metric for evaluation

Regarding the evaluation metric used in this paper, we follow DCASE 2021 Task 1A challenge. Let us consider C as the number of audio/visual test samples which are correctly classified, and the total number of audio/visual test samples is T, the classification accuracy (Acc. %) is the % ratio of C to T.

D. Optimize the proposed framework by evaluating factors of time length and data augmentation

In this paper, we further evaluate factors of time length and data augmentation which may affect the ASC performance, then find the most optimized framework. In particular, we evaluate four different time lengths of 1 second (i.e. 1-second patch is used in the baseline proposed), 2 seconds, 5 seconds and 10 seconds (i.e. an entire audio recording is 10-second length). In order to evaluate different time lengths mentioned but still keep the input patch of 128×128×3 unchanged, the hop size is set to 620, 1120, 1850 for 1-second, 2-second, and 5-second lengths respectively, while the other setting mentioned in Section II-A are unchanged. To evaluate an entire 10-second recording, the hop size is set to 2048, then generate one patch of 128×200×3.

We enforce the back-end classifiers by applying two methods of data augmentation on the patches: mixup [36], [37], and spectrum augmentation [38]. We then compare the ASC performance with and without using these data augmentation methods. As we apply mixup data augmentation [36], [37], the labels of the mixup data input are no longer one-hot. We therefore train back-end classifiers with Kullback-Leibler (KL) divergence loss [39] rather than the standard cross-entropy loss over all N mixup training samples:

$$LOSS_{KL}(\Theta) = \sum_{n=1}^{N} \mathbf{y}_n \log\left(\frac{\mathbf{y}_n}{\hat{\mathbf{y}}_n}\right) + \frac{\lambda}{2}||\Theta||_2^2, \qquad (5)$$

where Θ denotes the trainable network parameters and λ denote the ℓ_2-norm regularization coefficient. $\mathbf{y_c}$ and $\hat{\mathbf{y}}_\mathbf{c}$ denote the ground-truth and the network output, respectively. The training is carried out for 100 epochs using Adam [34] for optimization.

IV. Experimental Results And Discussion

A. Performance comparison between DCASE baseline and the CNN-7 baseline with or without using model compression methods

As experimental results are shown in Table IV, although the CNN-7 baseline outperforms DCASE baseline and helps to improve the accuracy by 12%, the CNN-7 baseline complexity is much larger than DCASE baseline. By using model compression methods, we can achieve a low-complexity model referred to as CNN-7 with CR & DC, which is nearly 1/2 of the DCASE baseline complexity, but still outperforms DCASE baseline, showing an accuracy improvement of 9%.

B. Effect of time length, data augmentation, spectrogram input

Next, we conduct experiments to evaluate effect of spectrogram input, time length, and data augmentations on the CNN-7 with CR & DC, which are shown in Table III. As experimental results are shown in Table III, MEL and GAM outperform CQT at different time lengths and with or without using data augmentation. Without using data augmentations, MEL obtains better results than GAM at different time lengths. However, both MEL and GAM achieve competitive results with using data augmentations. It can be seen that using data augmentations is effective only for GAM. Additionally, increasing time length helps to improve the accuracy for both CQT and MEL, but not much effective for GAM. As a result, we finally configure an optimized and low-complexity framework for ASC task with setting: CNN-7 with CR & DC, 5-second time length, and using mixup & spectrum data augmentations.

C. Evaluate ensemble of different spectrogram inputs

Given the optimized framework, we conduct PROD fusion of three predicted probabilities from three spectrogram inputs (i.e. PROD fusion is mentioned in Section II-B) to obtain the final classification accuracy. We then compare performances

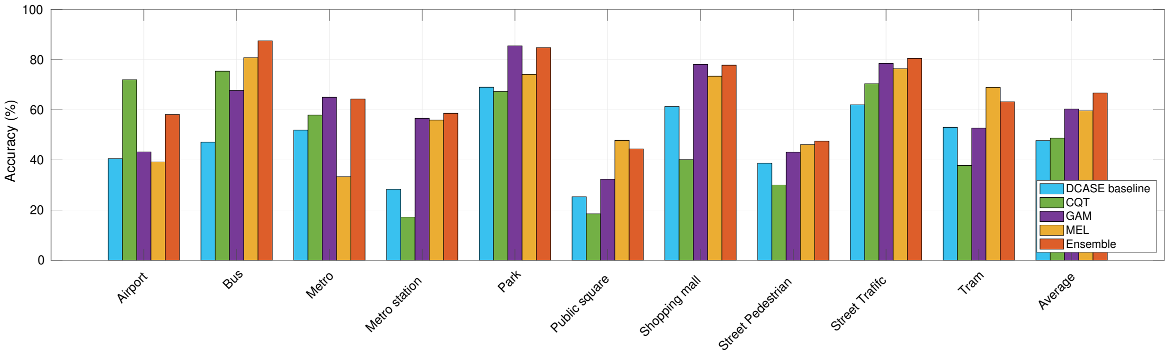


Fig. 3. Performance comparison (Acc.%) of DCASE baseline, individual spectrograms (CQT, GAM, and MEL), and the ensemble of three spectrograms across all scene categories (using CNN-7 with CR & DC, 5-second time length, and mixup & spectrum data augmentations)

TABLE V
THE NUMBER OF 10-SECOND AUDIO SCENE RECORDINGS CORRESPONDING TO EACH DEVICE IN THE TRAIN. AND EVAL. SUBSETS SEPARATED FROM THE DCASE 2021 TASK 1A DEVELOPMENT DATASET [35] AND PERFORMANCE FOR EACH DEVICES.

Devices	Train.	Eval.	Acc. %
A	10215	330	79.1
B	749	329	69.6
C	748	329	70.8
S1	750	330	65.8
S2	750	330	63.6
S3	750	330	67.0
S4	0	330	63.9
S5	0	330	60.0
S6	0	330	60.3

TABLE VI
TOP-10 ACCURACY PERFORMANCE (ACC. %) SYSTEMS SUBMITTED FOR DCASE 2021 TASK 1A CHALLENGE

Systems	Evaluation dataset	Eva. Subset
Top-1 [40]	76.1	75.9
Top-2 [41]	72.9	-
Top-3 [42]	72.1	69.5
Top-4 [43]	70.3	69.0
Top-5 [44]	70.1	-
Our system	**69.6**	**66.7**
Top-7 [45]	69.6	65.0
Top-8 [46]	68.8	70.2
Top-9 [47]	68.5	65.2
Top-10 [48]	68.3	69.7
DCASE baseline [31]	45.6	47.7

among DCASE baseline, the optimized framework with individual spectrograms, the optimized framework with the ensemble of multiple spectrograms, across all scene categories. As experimental results are shown in Fig.3, GAM and MEL achieve competitive results, and outperform CQT at almost scene categories except for 'Airport'. The ensemble of three spectrogram inputs helps to achieve an average accuracy of 66.7%, improving DCASE baseline by 19.0%, and notably showing improvement over all scene categories.

Further analysing performance over different recording devices as shown in Table V, we see that device A outperforms the other devices as this device is dominant in Train. subset. Although there is a lacking of training samples for device B and C, they achieves competitive accuracy of 69.6% and 70.8% respectively, compared with device A performance of 79.1%. Regarding synthesized devices from S1 to S6, although there is no samples from S4, S5, S6 in Train. subset, the performance of these devices are competitive to the other S1, S2, S3. Our results and analysis indicate that the ASC framework proposed not only achieves a low complexity of 128 KB, but it also can tackle the issue of mismatched recording devices.

D. Compare with the state-of-the-art systems

Compare with the state-of-the-art systems as shown in Table VI, our result on Evaluation dataset [32] achieves 69.6%, occupying top-6 team ranking with respect to accuracy performance. Furthermore, the low gap of accuracy performance between Eval. Subset [35] (66.7%) and Evaluation dataset [32] (69.6%) proves our proposed framework robust and general.

V. CONCLUSION

We have just presented a low-complexity framework for ASC, which makes use of multiple spectrogram inputs and model compression techniques. While the ensemble of multiple spectrograms helps to tackle different ASC challenges of mismatch recording devices or lacking input to improve the ASC performance, a combination of model restriction and decomposed convolution techniques is effective to achieve a low model complexity of 128 KB. In the future, we will further compress the model complexity by combining our proposed approaches with other techniques of distillation, pruning, and quantization.

ACKNOWLEDGEMENT

The AMMONIS project is funded by the FORTE program of the Austrian Research Promotion Agency (FFG) and the Federal Ministry of Agriculture, Regions and Tourism (BMLRT) under grant no. 879705.

REFERENCES

[1] R. F. Lyon, *Human and Machine Hearing*. Cambridge University Press, 2017.

[2] S. Ravindran and D. Anderson, "Audio classification and scene recognition and for hearing aids," in *IEEE International Symposium on Circuits and Systems*, 2005, pp. 860–863.

[3] J. Xiang, M. F. McKinney, K. Fitz, and T. Zhang, "Evaluation of sound classification algorithms for hearing aid applications," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2010, pp. 185–188.

[4] D. Oldoni, B. De Coensel, M. Rademaker, B. De Baets, and D. Botteldooren, "Context-dependent environmental sound monitoring using som coupled with legion," in *The International Joint Conference on Neural Networks (IJCNN)*, 2010, pp. 1–8.

[5] L. Yang, X. Chen, and L. Tao, "Acoustic scene classification using multi-scale features," in *Proc. DCASE*, 2018, pp. 29–33.

[6] L. Pham, I. Mcloughlin, H. Phan, R. Palaniappan, and A. Mertins, "Deep feature embedding and hierarchical classification for audio scene classification," in *International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–7.

[7] D. Ngo, H. Hoang, A. Nguyen, T. Ly, and L. Pham, "Sound context classification basing on join learning model and multi-spectrogram features," *ArXiv*, vol. abs/2005.12779, 2020.

[8] T. Nguyen and F. Pernkopf, "Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters," in *Proc. DCASE*, 2018, pp. 34–38.

[9] H. Phan, H. Le Nguyen, O. Y. Chén, L. Pham, P. Koch, I. McLoughlin, and A. Mertins, "Multi-view audio and music classification," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 611–615.

[10] Y. Sakashita and M. Aono, "Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions," DCASE Challenge, Tech. Rep., 2018.

[11] H. Phan, O. Y. Chén, P. Koch, L. Pham, I. McLoughlin, A. Mertins, and M. De Vos, "Beyond equal-length snippets: How long is sufficient to recognize an audio scene?" in *Proc. AES*, Jun 2019.

[12] H. Phan, O. Chén, L. Pham, P. Koch, M. de Vos, I. Mcloughlin, and A. Mertins, "Spatio-temporal attention pooling for audio scene classification," in *Proc. INTERSPEECH*, 2019, pp. 3845–3849.

[13] L. Pham, H. Phan, T. Nguyen, R. Palaniappan, A. Mertins, and I. Mcloughlin, "Robust acoustic scene classification using a multi-spectrogram encoder-decoder framework," *Digital Signal Processing*, vol. 110, p. 102943, 2021.

[14] S. Phaye, E. Benetos, and Y. Wang, "SubSpectralNet using sub-spectrogram based convolutional neural networks for acoustic scene classification," in *Proc. ICASSP*, 2019, pp. 825–829.

[15] Z. Ren, K. Qian, Y. Wang, Z. Zhang, V. Pandit, A. Baird, and B. Schuller, "Deep scalogram representations for acoustic scene classification," *IEEE/CAA Journal of Automatica Sinica*, vol. 5, no. 3, pp. 662–669, 2018.

[16] L. Pham, I. McLoughlin, H. Phan, R. Palaniappan, and Y. Lang, "Bag-of-features models based on C-DNN network for acoustic scene classification," in *Proc. AES*, 2019.

[17] L. Pham, I. Mcloughlin, H. Phan, and R. Palaniappan, "A robust framework for acoustic scene classification," in *Proc. INTERSPEECH*, 2019, pp. 3634–3638.

[18] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, "Efficient processing of deep neural networks: A tutorial and survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017.

[19] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," *arXiv preprint arXiv:1506.02626*, 2015.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[21] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[22] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 6105–6114.

[23] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," *arXiv preprint arXiv:1511.06530*, 2015.

[24] V. Joseph, S. A. Siddiqui, A. Bhaskara, G. Gopalakrishnan, S. Muralidharan, M. Garland, S. Ahmed, and A. Dengel, "Reliable model compression via label-preservation-aware loss functions," *arXiv preprint arXiv:2012.01604*, 2020.

[25] B. McFee, R. Colin, L. Dawen, D. Ellis, M. Matt, B. Eric, and N. Oriol, "librosa: Audio and music signal analysis in python," in *Proceedings of The 14th Python in Science Conference*, 2015, pp. 18–25.

[26] D. P. W. . Ellis, "Gammatone-like spectrogram," 2009. [Online]. Available: http://www.ee.columbia.edu/ dpwe/resources/matlab/ gammatonegram

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 448–456.

[28] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *ICML*, 2010.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] K. Koutini, F. Henkel, H. Eghbal-zadeh, and G. Widmer, "Low-complexity models for acoustic scene classification based on receptive field regularization and frequency damping," *arXiv preprint arXiv:2011.02955*, 2020.

[31] A. Mesaros, T. Heittola, and T. Virtanen, "A multi-device dataset for urban acoustic scene classification," in *Proc. DCASE*, 2018, pp. 9–13.

[32] D. community, *TAU Urban Acoustic Scenes 2021 Mobile, Evaluation dataset*, https://zenodo.org/record/4767109.YTCOUpozZhE.

[33] Detection and Classification of Acoustic Scenes and Events Community, *DCASE 2021 challenges*, http://dcase.community/challenge2021.

[34] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

[35] S. Wang, A. Mesaros, T. Heittola, and T. Virtanen, "A curated dataset of urban scenes for audio-visual scene analysis," *arXiv preprint arXiv:2011.00030*, 2020.

[36] K. Xu, D. Feng, H. Mi, B. Zhu, D. Wang, L. Zhang, H. Cai, and S. Liu, "Mixup-based acoustic scene classification using multi-channel convolutional neural network," in *Pacific Rim Conference on Multimedia*, 2018, pp. 14–23.

[37] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from between-class examples for deep sound recognition," in *ICLR*, 2018.

[38] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[39] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[40] B. Kim, S. Yang, J. Kim, and S. Chang, "QTI submission to DCASE 2021: Residual normalization for device-imbalanced acoustic scene classification with efficient design," DCASE2021 Challenge, Tech. Rep., June 2021.

[41] C.-H. H. Yang, H. Hu, S. M. Siniscalchi, Q. Wang, W. Yuyang, X. Xia, Y. Zhao, Y. Wu, Y. Wang, J. Du, and C.-H. Lee, "A lottery ticket hypothesis framework for low-complexity device-robust neural acoustic scene classification," DCASE2021 Challenge, Tech. Rep., June 2021.

[42] K. Koutini, S. Jan, and G. Widmer, "Cpjku submission to dcase21: Cross-device audio scene classification with wide sparse frequency-damped CNNs," DCASE2021 Challenge, Tech. Rep., June 2021.

[43] S. Seo and J.-H. Kim, "Mobilenet using coordinate attention and fusions for low-complexity acoustic scene classification with multiple devices," DCASE2021 Challenge, Tech. Rep., June 2021.

[44] H. Hee-Soo, J. Jee-weon, S. Hye-jin, and L. Bong-Jin, "Clova submission for the DCASE 2021 challenge: Acoustic scene classification using light architectures and device augmentation," DCASE2021 Challenge, Tech. Rep., June 2021.

[45] Y. Liu, J. Liang, L. Zhao, J. Liu, K. Zhao, W. Liu, L. Zhang, T. Xu, and C. Shi, "DCASE 2021 task 1 subtask a: Low-complexity acoustic scene classification," DCASE2021 Challenge, Tech. Rep., June 2021.

[46] L. Byttebier, B. Desplanques, J. Thienpondt, S. Song, K. Demuynck, and N. Madhu, "Small-footprint acoustic scene classification through 8-bit quantization-aware training and pruning of ResNet models," DCASE2021 Challenge, Tech. Rep., June 2021.

[47] S. Lim, Y. Lee, and I.-Y. Kwak, "CAU-ET submission to DCASE 2021: Light-efficientnet for acoustic scene classification," DCASE2021 Challenge, Tech. Rep., June 2021.

[48] M. Cui, F. Kui, and L. Guo, "Consistency learning based acoustic scene classification with res-attention," DCASE2021 Challenge, Tech. Rep., June 2021.

Anomaly Detection in Medical Imaging - A Mini Review

Maximilian E. Tschuchnig and Michael Gadermayr
Salzburg University of Applied Sciences, 5412 Puch, Austria
{maximilian.tschuchnig, michael.gadermayr}@fh-salzburg.ac.at

***Abstract*—The increasing digitization of medical imaging enables machine learning based improvements in detecting, visualizing and segmenting lesions, easing the workload for medical experts. However, supervised machine learning requires reliable labelled data, which is is often difficult or impossible to collect or at least time consuming and thereby costly. Therefore methods requiring only partly labeled data (semi-supervised) or no labeling at all (unsupervised methods) have been applied more regularly. Anomaly detection is one possible methodology that is able to leverage semi-supervised and unsupervised methods to handle medical imaging tasks like classification and segmentation. This paper uses a semi-exhaustive literature review of relevant anomaly detection papers in medical imaging to cluster into applications, highlight important results, establish lessons learned and give further advice on how to approach anomaly detection in medical imaging. The qualitative analysis is based on google scholar and 4 different search terms, resulting in 120 different analysed papers. The main results showed that the current research is mostly motivated by reducing the need for labelled data. Also, the successful and substantial amount of research in the brain MRI domain shows the potential for applications in further domains like OCT and chest X-ray.**

***Index Terms*—anomaly detection, medical imaging, lessons learned**

I. Introduction

The increasing digitization of medical imaging enables the collection of data and machine learning (ML) based approaches to aid medical experts. One powerful part of ML comes from supervised methods, using both data and corresponding labels in e.g. segmentation or classification models. However, since the collection of annotations (labels) is often times time consuming and thereby costly [1] as well as in many cases a confident ground truth even being unobtainable, their usability is reduced. Due to this, semi-supervised and unsupervised methods are applied. This is often achieved through anomaly detection.

Definitions: Pathologies in medical images can often be described as a rare deviance from a norm, or a non-anomalous (in the case of medical imaging mostly healthy) sample. This fits the definition of outliers (or anomalies) in the data, motivating the application of anomaly detection [2]. In this publication, the terms anomaly detection and outlier detection are used interchangeably. This is motivated by the fact that outliers are sometimes defined as valid but out of order datapoints, while anomalies also include further differences (e.g. different image capture modalities). Therefore outliers can be defined as a subset of anomalies. Anomaly detection can be separated into 3 classes, *point*, *collective* and *contextual* anomalies. Point anomaly detection is the task of recognizing a single anomalous point from a larger dataset. Most anomaly detection models handle point anomalies. Collective anomalies are anomalies that may not be identified as anomalies if viewed as a single point but as a set of many they form an anomaly. Contextual anomalies can only be recognized as anomalies if context is added. There are also 3 different anomaly detection setups, *supervised*, *semi-supervised* and *unsupervised* anomaly detection. Supervised anomaly detection is comparable with classification using a very unbalanced dataset. Semi-supervised anomaly detection aims to train a model on only one, typically the normal (in our case healthy) class and then applies the model to both healthy and pathological data, reporting the corresponding scores. Unsupervised anomaly detection uses both, normal and anomalous data, does not make use of labels at all and works purely on intrinsic properties of the dataset (using distances or densities) [3]. In anomaly detection, the usage of semi-supervised and unsupervised anomaly detection (UAD) is confused, and repeatedly applied to both semi-supervised and unsupervised methods. We believe that the separation into semi-supervised (healthy data being clearly defined) and unsupervised (no definition of labels at all) makes sense and advise to use this terminology as also pointed out by [3].

Deviation based anomaly detection: Anomaly detection using medical image data, e.g. computed tomography (CT) scans, is typically performed using either convolutional neural network (CNN) based feature extractors, followed by one-class (OC) classifiers or deviation based methods like autoencoders (AEs) [4]–[6] or even more recently, generative adversarial network (GAN) [7]–[9] based methods. Both AEs and GANs use convolutional kernels, however their applications in the sense of deviation based anomaly detection are fundamentally different to CNN based feature extractors. In order to generate deviation based scores from an AE, the encoder of the encoder-decoder based neural network typically encodes a sample image into a lower dimensional latent space, also called a bottleneck. The decoder uses this latent space representation to recreate the sample and a deviation between the sample and the reconstruction can be calculated. During training, this deviation is used to backpropagate and update the network. The AE in an anomaly detection setting is trained using healthy data to en- and decode features of healthy samples, leading to a higher deviation for non-healthy

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_5

samples, assuming that there is a difference between the learned healthy and the lesioned latent space [10]. GANs can also be used to facilitate a deviation based score. In addition to training a generator and a discriminator in an adversarial setup, an additional encoder needs to be trained, mapping the generated image back to the latent space (input to generator) [7]. By doing this, any input image can be mapped to a latent space and reconstructed into an image using the generator. This results in a reconstruction which can be used to facilitate a reconstruction loss.

Additionally there are conventional methods to facilitate anomaly detection, using e.g. z-score thresholds [11], [12], boxplots [13] or methods built on the ideas of principal components analysis (PCA) [14], [15]. OC support vector machines (SVM)s [16] are one of the most known semi-supervised anomaly detection methods. In principle they apply the ideas of SVMs (using hyperplanes to separate two classes using support vectors with the aim of generating the largest possible margin) to a OC problem. One possibility to achieve this is to model a hypersphere to encompass all support vectors, creating the smallest possible hypersphere.

Contribution: This papers contribution is the analysis of the current state of anomaly detection in medical imaging. Using this analysis, we show lessons learned and give an outlook for future applications and research targets.

II. Method

The method used was a semi-exhaustive literature review based on Randolph [17]. The formulated problem was the evaluation of anomaly detection in medical imaging. For data collection, the search engine Google Scholar was used. In order to obtain meaningful results, the search terms *anomaly detection in medical imaging*, *unsupervised anomaly detection in medical imaging*, *outlier detection in medical imaging* and *unsupervised outlier detection in medical imaging* were chosen. From these results the following criteria for exclusion were chosen. Only the first 3 pages of results (sorted by relevance, 10 articles per page) were used. Further, the criteria for exclusion *duplicate*, *in context of medical imaging (in abstract, title or conclusion)*, *peer-review* and *date* were identified. Since the search terms were similar, *duplicates* had to be removed. Papers without a clear focus on *medical imaging* in the abstract, title or conclusion were also removed. A further criterion was to only include *peer-reviewed* research items. This mainly lead to the exclusion of preprints. The data timeframe was set to not include papers after the resurgence of deep-learning (AlexNet [18]) and to still include papers after the U-net was proposed [19], resulting in a timeframe of January 2015 – July 2021. This lead to a reduction of papers from 120 to 49. Since these papers also included 4 survey papers, the final number of application based research papers was 45. These survey papers were used as a qualitative comparision to the our extracted lessons learned. Next, the papers were manually clustered with respect to their imaging method and the following information was extracted: *Aim*, *Applied Method* and *Results*. From these clusters, lessons learned were extracted, which are reported in section III.

III. Results

The semi-exhaustive literature review resulted in 45 research items, from which further 6 were removed due to not containing applications in medical imaging (only exemplar stated in abstract) or being non-available. The resulting papers were further clustered into 5 categories (corresponding to Tab. I-V by their imaging methods. Tab. I shows papers applying anomaly detection to occular medical images with retinal fundus images and optical coherence tomography (OCT). Tab. II focuses on papers with applications in the center body region, with chest X-rays and mammography. Tab. III summarizes application papers, using CT and functional magnetic resonance imaging (fMRI). Tab. IV displays papers applying ML to brain Magnetic resonance imaging (MRI) datasets. Tab. V shows mixed applications from the domains of breast ultrasound, chest radiographs, histology and fundus images as well as multi-spectral imaging (MSI).

Overall, these tables show a narrow field of application with 15 (38.46%) of all selected papers working on MRI scans of the brain. Further 6 papers use fMRI and CT scans of the brain, increasing the amount of brain image data based applications to 53.85%. Further clusters could be observed using chest X-rays and mammography, as well as ocular imaging techniques, especially OCT. Of note is, that although medical imaging includes methods like histology, only 1 paper [20] applied anomaly detection to such data. A further result is the relevance of deviation based methods, with 27 papers (69.23%) applying some form or adaptation, mostly using autoencoders AEs or GANs [7]–[9], [20]–[43]. Investigating MRIs, 7 [37], [39], [40], [42]–[45] of the 15 publications using brain MRI data focus explicitly on tumours or metastases, showing the usefulness of anomaly detection and segmentation of tumours in brains using MRI. Most other brain MRI based methods more generally handle the task of lesion classification or segmentation with only two focusing specificly on cerebral small vessel diseases [41], [46]. A further cluster uses X-ray for the detection of pneumonia [23], [47] or lung disease like COVID-19 [15]. Several advancements have also been made in OCT segmentation of retina lesions [7], [8], [25], [26], [48], with one publication performing visual touring test using 2 experts, which were unable to recognize differences in the correctly reconstructed data [8]. Breast cancer and pathology detection was also improved using anomaly detection [28], [29], [49].

One result of this analysis is the statement that anomaly detection can be motivated by the lack of available labelled training data, which was stated in 19 publications. The reported results of these papers proved that these semi- and unsupervised approaches successfully completed their tasks [7], [8], [14], [20], [22], [24]–[26], [28], [29], [31], [33], [36], [38], [40], [41], [44], [48], [49]. However, some papers also show semi-supervised methods outperforming fully supervised methods. These outperforming methods are

TABLE I
TABLE CONSISTING OF OCULAR IMAGE BASED RESULTS OBTAINED BY THE LITERATURE REVIEW

Paper	Imaging Method	Aim	Applied Method
[48]	retinal fundus images	transfer learning (general and retinal lesions)	TL (IMNet feature extrator)
[7]	OCT	new anomaly detection method	AnoGAN
[8]	OCT	new anomaly detection method	fAno-GAN
[25]	OCT	segmentation (retina lesions)	Bayesian U-Net. Episdemic uncertainty estimations and post processing
[26]	OCT and chest X-ray	new anomaly detection method	encoder-decoder with additional GAN discriminators

TABLE II
TABLE CONSISTING OF CENTER BODY IMAGE BASED RESULTS OBTAINED BY THE LITERATURE REVIEW

Paper	Imaging Method	Aim	Applied Method
[23]	chest radiographs	anomaly detection (pneumonia)	α-GAN
[47]	chest X-ray	anomaly detection (virial pneumonia)	CNN feature extractor with anomaly score (Fully connected) and confidency (Fully connected)
[27]	chest X-ray	new anomaly detection method (pleural effusions)	DeScarGAN
[15]	chest X-ray	anomaly detection (coronavirus)	edge detection and morphology. PCA to reduce features and use in RNN
[28]	mammography	anomaly detection (compressions or implants)	Stacked AE as feature extractor, K-Means for clustering
[49]	(MIL) mammography	anomaly detection (breast cancer)	Simultaniously trained MIL algorithms (DD, APR, and MIL-Boost)
[29]	mammography	anomaly detection (breast anomalies)	cAE with RMSD threshold

TABLE III
TABLE CONSISTING OF CT AND fMRI IMAGE BASED RESULTS OF THE BRAIN OBTAINED BY THE LITERATURE REVIEW

Paper	Imaging Method	Aim	Applied Method
[21]	head CT (3D)	anomaly detection (emergency head CTs)	3D cAE
[30]	brain CT (2D)	anomaly detection (brain lesions)	Bayesian AE
[31]	PET-CT and brain MRI	image-to-image translation (image artifacts)	Cycle-MedGAN
[14]	Brain fMRI	pca based outlier removal (image artifacts)	PCA (robust distance and leverage)
[32]	brain rs-fMRI		AE and frame prediction (conv-LSTM)
[50]	Brain fMRI	anomaly detection using constraint programming (cognitive impairment)	Constraint Programming using 3 constraints

based on classical feature extraction followed by multiple-instance learning (MIL) based models [49], through adaptations to GANs [27] (using skip-connections and weight-sharing subnetworks) and through the adaptation of AEs to the SegAE model [32] (using pairs of T1-w, T2-w and FLAIR data for improved anomaly detection). For this improvement in comparison to fully supervised models, Khosla et al. [32] reason that fully supervised methods systematically either under or overestimate lesion volumes (when segmenting lesions), while their proposed method was reported to be free of this bias.

Zhang et al. and Kim et al. both show interesting approaches, applying conventional feature extractors (CNN and edge detection) with further OC classifiers (fully connected neural networks and recurrent neural networks). By using these methods both papers reach relatively high scores, but still lower scores then their CT based baselines.

A further finding is the obvious bias in the amount of research items regarding OCT, chest X-Ray, mammography and Brain MRI. An investigation in the used datasets shows a strong dataset and community driven effect. For all of the above mentioned image categories, datasets are publicly available. Further, a community driven effect can be observed, comparing new models against older ones, evaluated on the same dataset.

In addition to medical image based application papers, several authors proposed improvements to the general anomaly detection pipelines. 3 papers showed an improvement of subsequent methods by removing anomalies from the data or reducing complexity in the data [11], [12], [14]. Also, constraint programming is shown successfully by Kuo et al. [50]. showing further approaches to perform anomaly detection. CycleGAN is also shown to work for transforming images into a space that showed reduced image artefacts [31]. Heer et al. [38] showed issues with the general idea of anomaly detection and their application of anomalies as out-of-distribution (OOD) data, remarking a blind spot using deviation based methods. They state that denoting anomalies as OOD is

TABLE IV
TABLE CONSISTING OF FMRI IMAGE BASED RESULTS OF THE BRAIN OBTAINED BY THE LITERATURE REVIEW

Paper	Imaging Method	Aim	Applied Method
[33]	brain MRI	segmentation (brain lesions)	SegAE
[34]	brain MRI	anomaly detection (epilepsy)	siamese network, stacked cAE, wasserstein AE
[35]	brain MRI	improvements to AE based methods (glioma)	VAE + LG (and several baselines)
[46]	brain MRI	segmentation (cerebral small vessel disease)	PHI-Syn [51] (image synthesis) and Gaussian mixture models used by oc-SVM
[44]	Brain MRI	segmentation (brain lesions)	Hidden markov models
[36]	Brain MRI	anomaly detection (brain lesion)	siamese, stacked cAE for latent representations in oc-SVM
[45]	Brain MRI	segmentation (brain tumor)	DistGP-Seg. Incooperating DistGP into CNN
[37]	Brain MRI	anomaly detection (MS and cancer)	spatial AE with skip connections
[38]	Brain MRI	awareness for OOD	VAE. Scores: l1, Kullback–Leibler divergence, Watanabe–Akaike information criterion score, Density of States Estimation
[39]	Brain MRI	improvements to cycleGAN (brain tumor)	SteGANomaly
[9]	Brain MRI	anomaly segmentation (brain lesions)	AnoVAEGan
[40]	Brain MRI	anomal localization (brain tumor)	VAE with additional KL divergence term in Backprop
[41]	Brain MRI	anomaly detection (brain infarct)	GANomaly
[42]	brain MRI	new anomaly detection method (brain mestastases)	(Wasserstein based) MaDGAN using self attention (paired)
[13]	Brain MRI (DTI)	quality assurance of segmentation (brain lesions)	non parametric (box-plots); supervised classification models
[43]	Brain MRI	new anomaly detection method (tumor)	GMVAE

TABLE V
TABLE CONSISTING OF REMAINING MIXED LITERATURE REVIEW RESULTS

Paper	Imaging Method	Aim	Applied Method
[22]	breast ultrasound	anomaly detection (normal, begning, malignant in breasts)	bidirectional GAN
[20]	hisotlogy images	image synthesis (tumor)	DCGAN & WGAN
[24]	fundus image	anomly localization (glaucoma)	adversarial attention guided VAE
[11]	MSI	outlier removal to improve burn detection	z-score based outlier detection to improve SVM and KNN
[12]	MSI	outlier removal to improve burn detection	z-score based outlier detection to improve SVM and KNN

dangerous, since non anomalous data from different sensors or image modalities may also be detected as OOD although this data not being anomalous. In their paper they further present a method based on prior knowledge to disentangle lesion based OOD from non-lesion based counterparts.

IV. DISCUSSION

In this paper we analysed the current state of research in anomaly detection using medical image data and extracted lessons learned. To accomplish this, a semi-exhaustive literature review was performed, resulting in 120 papers, from which 44 were further investigated (after filters were applied). This resulted in 4 major clusters of image domains, with the brain MRI domain comprising 39.45% of all papers.

One takeway is that especially in the brain MRI domain, both lesion and tumour classification as well as segmentation have been successfully implemented multiple times. It is shown that both AE and GAN based methods as well as Gaussian mixture models, hidden Markov models and CNNs with specific feature extractors can work in this anomaly detection setup. This was further shown to be the case with chest X-ray, mammography as well as OCT data. Extrapolating from these results, first approaches in similar domains, using anomaly detection for tasks in the domains of e.g. CT scans of the skeleton or spines seem promising and should be investigated. Also, an investigation of the suitability for histological data would be of high interest, since histological data was very under-represented (1 publication). However, there are multiple differences between CT/MRI and histology. In histology it is not sufficient to detect a large object (e.g. tumor) which is indicated by different intensity values. It would rather be important to learn the shape and interaction of nuclei and cells which is supposed to be a more challenging task, relying more on high frequency information which is a reported weak point of several proposed deviation based mehtods. Further, histology images are extremely high resolution, leading to issues using current GAN or AE based anomaly detection. Štepec et al. [20] show one way to circumvent these issues successfully using patch extraction and MIL.

As reported in the results, there were some semi-supervised anomaly detection models that resulted in higher or similar scores than their fully supervised alternatives. One interpretation is that, especially regarding segmentation, human labelled segmentation masks with rough edges may introduce bias. This is however still unclear and should further be investigated.

Another useful takeaway is that not only improvements to state of the art (SOTA) models are needed but also simpler models or cheaper image modalities can be a major improvement, even if the SOTA scores cannot be reached e.g. replacing CT with X-ray based methods. One example was shown by Zhang et al. [47] who used X-ray images, approaching relatively high scores. Although their method did not outperform the CT based baselines, the methods is still of high significance, since it reached similar levels using X-rays requiring a lower radiation dose and an more available imaging method.

As stated by [52] we also recognized the generation of free and comparable datasets as a high priority to facilitate further research. The fast growing brain MRI community showed, that open datasets are an important asset to boost research. Therefore the development and open distribution should be pursued for different medical image domains. In order to facilitate anomaly detection research, a semi-supervised dataset (only including a small amount of annotations) should be developed.

A disadvantage, reported by several deep learning based approaches was [30], [44], that results were still unstable and more research was needed before a clinical application could be performed. This however was not always the case [22] but there are still doubts in the clinical applicability of deep learning based anomaly detection methods. Large clinical application studies would be needed to show their suitability.

Conclusion: In this paper we investigated the current state of research in medical image based anomaly detection and generated lessons learned. The lessons learned can be converted into the following future targets: *a very narrow domain of application that should be expanded*, *development of freely accessible datasets*, *investigation of the OCT blindspot* and *improvements of working approaches like constraints on the AE bottleneck*.

Acknowledgment

This work was partially funded by the County of Salzburg under grant number FHS-2019-10-KIAMed.

References

[1] R. Domingues, M. Filippone, P. Michiardi, and J. Zouaoui, "A comparative evaluation of outlier detection algorithms: Experiments and analyses," *Pattern Recognition*, vol. 74, pp. 406–421, 2018.

[2] F. E. Grubbs, "Procedures for detecting outlying observations in samples," *Technometrics*, vol. 11, no. 1, pp. 1–21, 1969.

[3] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PloS one*, vol. 11, no. 4, p. e0152173, 2016.

[4] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.

[5] J. Sun, X. Wang, N. Xiong, and J. Shao, "Learning sparse representation with variational auto-encoder for anomaly detection," pp. 33 353–33 361, 2018.

[6] H. Uzunova, S. Schultz, H. Handels, and J. Ehrhardt, "Unsupervised pathology detection in medical images using conditional variational autoencoders," *International journal of computer assisted radiology and surgery*, vol. 14, no. 3, pp. 451–461, 2019.

[7] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International conference on information processing in medical imaging*. Springer, 2017, pp. 146–157.

[8] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-anogan: Fast unsupervised anomaly detection with generative adversarial networks," *Medical image analysis*, vol. 54, pp. 30–44, 2019.

[9] C. Baur, B. Wiestler, S. Albarqouni, and N. Navab, "Deep autoencoding models for unsupervised anomaly segmentation in brain mr images," in *International MICCAI Brainlesion Workshop*. Springer, 2018, pp. 161–169.

[10] D. Gong, L. Liu, V. Le, B. Saha, M. R. Mansour, S. Venkatesh, and A. v. d. Hengel, "Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1705–1714.

[11] W. Li, W. Mo, X. Zhang, J. J. Squiers, Y. Lu, E. W. Sellke, W. Fan, J. M. DiMaio, and J. E. Thatcher, "Outlier detection and removal improves accuracy of machine learning approach to multispectral burn diagnostic imaging," *Journal of biomedical optics*, vol. 20, no. 12, p. 121305, 2015.

[12] W. Li, W. Mo, X. Zhang, Y. Lu, J. J. Squiers, E. W. Sellke, W. Fan, J. M. DiMaio, and J. E. Thatcher, "Burn injury diagnostic imaging device's accuracy improved by outlier detection and removal," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XXI*, vol. 9472. International Society for Optics and Photonics, 2015, p. 947206.

[13] K. Li, C. Ye, Z. Yang, A. Carass, S. H. Ying, and J. L. Prince, "Quality assurance using outlier detection on an automatic segmentation method for the cerebellar peduncles," in *Medical Imaging 2016: Image Processing*, vol. 9784. International Society for Optics and Photonics, 2016, p. 97841H.

[14] A. F. Mejia, M. B. Nebel, A. Eloyan, B. Caffo, and M. A. Lindquist, "Pca leverage: outlier detection for high-dimensional functional magnetic resonance imaging data," *Biostatistics*, vol. 18, no. 3, pp. 521–536, 2017.

[15] C.-M. Kim, E. J. Hong, and R. C. Park, "Chest x-ray outlier detection model using dimension reduction and edge detection," *IEEE Access*, 2021.

[16] D. M. Tax and R. P. Duin, "Uniform object generation for optimizing one-class classifiers," *Journal of machine learning research*, vol. 2, no. Dec, pp. 155–173, 2001.

[17] J. Randolph, "A guide to writing the dissertation literature review," *Practical Assessment, Research, and Evaluation*, vol. 14, no. 1, p. 13, 2009.

[18] A. Krizhevsky, I. Sutskever, and G. Hinton, "2012 alexnet," pp. 1–9, 2012.

[19] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[20] D. Štepec and D. Skočaj, "Image synthesis as a pretext for unsupervised histopathological diagnosis," in *International Workshop on Simulation and Synthesis in Medical Imaging*. Springer, 2020, pp. 174–183.

[21] D. Sato, S. Hanaoka, Y. Nomura, T. Takenaga, S. Miki, T. Yoshikawa, N. Hayashi, and O. Abe, "A primitive study on unsupervised anomaly detection with an autoencoder in emergency head ct volumes," in *Medical Imaging 2018: Computer-Aided Diagnosis*, vol. 10575. International Society for Optics and Photonics, 2018, p. 105751P.

[22] T. Fujioka, K. Kubota, M. Mori, Y. Kikuchi, L. Katsuta, M. Kimura, E. Yamaga, M. Adachi, G. Oda, T. Nakagawa *et al.*, "Efficient anomaly detection with generative adversarial network for breast ultrasound imaging," *Diagnostics*, vol. 10, no. 7, p. 456, 2020.

[23] T. Nakao, S. Hanaoka, Y. Nomura, M. Murata, T. Takenaga, S. Miki, T. Watadani, T. Yoshikawa, N. Hayashi, and O. Abe, "Unsupervised deep anomaly detection in chest radiographs," *Journal of Digital Imaging*, pp. 1–10, 2021.

[24] S. Venkataramanan, K.-C. Peng, R. V. Singh, and A. Mahalanobis, "Attention guided anomaly localization in images," in *European Conference on Computer Vision*. Springer, 2020, pp. 485–503.

[25] P. Seeböck, J. I. Orlando, T. Schlegl, S. M. Waldstein, H. Bogunović, S. Klimscha, G. Langs, and U. Schmidt-Erfurth, "Exploiting epistemic uncertainty of anatomy segmentation for anomaly detection in retinal oct," *IEEE transactions on medical imaging*, vol. 39, no. 1, pp. 87–98, 2019.

[26] H. Zhao, Y. Li, N. He, K. Ma, L. Fang, H. Li, and Y. Zheng, "Anomaly detection for medical images using self-supervised and translation-consistent features," *IEEE Transactions on Medical Imaging*, 2021.

[27] J. Wolleb, R. Sandkühler, and P. C. Cattin, "Descargan: Disease-specific anomaly detection with weak supervision," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 14–24.

[28] T. Tlusty, G. Amit, and R. Ben-Ari, "Unsupervised clustering of mammograms for outlier detection and breast density estimation," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3808–3813.

[29] Q. Wei, Y. Ren, R. Hou, B. Shi, J. Y. Lo, and L. Carin, "Anomaly detection for medical images based on a one-class classification," in *Medical Imaging 2018: Computer-Aided Diagnosis*, vol. 10575. International Society for Optics and Photonics, 2018, p. 105751M.

[30] N. Pawlowski, M. C. Lee, M. Rajchl, S. McDonagh, E. Ferrante, K. Kamnitsas, S. Cooke, S. Stevenson, A. Khetani, T. Newman *et al.*, "Unsupervised lesion detection in brain ct using bayesian convolutional autoencoders," 2018.

[31] K. Armanious, C. Jiang, S. Abdulatif, T. Küstner, S. Gatidis, and B. Yang, "Unsupervised medical image translation using cycle-medgan," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.

[32] M. Khosla, K. Jamison, A. Kuceyeski, and M. R. Sabuncu, "Detecting abnormalities in resting-state dynamics: An unsupervised learning approach," in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2019, pp. 301–309.

[33] H. E. Atlason, A. Love, S. Sigurdsson, V. Gudnason, and L. M. Ellingsen, "Unsupervised brain lesion segmentation from mri using a convolutional autoencoder," in *Medical Imaging 2019: Image Processing*, vol. 10949. International Society for Optics and Photonics, 2019, p. 109491H.

[34] Z. Alaverdyan, J. Chai, and C. Lartizien, "Unsupervised feature learning for outlier detection with stacked convolutional autoencoders, siamese networks and wasserstein autoencoders: application to epilepsy detection," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Springer, 2018, pp. 210–217.

[35] X. Chen, N. Pawlowski, B. Glocker, and E. Konukoglu, "Unsupervised lesion detection with locally gaussian approximation," in *International Workshop on Machine Learning in Medical Imaging*. Springer, 2019, pp. 355–363.

[36] Z. Alaverdyan, J. Jung, R. Bouet, and C. Lartizien, "Regularized siamese neural network for unsupervised outlier detection on brain multiparametric magnetic resonance imaging: application to epilepsy lesion screening," *Medical image analysis*, vol. 60, p. 101618, 2020.

[37] C. Baur, B. Wiestler, M. Muehlau, C. Zimmer, N. Navab, and S. Albarqouni, "Modeling healthy anatomy with artificial intelligence for unsupervised anomaly detection in brain mri," *Radiology: Artificial Intelligence*, vol. 3, no. 3, p. e190169, 2021.

[38] M. Heer, J. Postels, X. Chen, E. Konukoglu, and S. Albarqouni, "The ood blind spot of unsupervised anomaly detection," in *Medical Imaging with Deep Learning*, 2021.

[39] C. Baur, R. Graf, B. Wiestler, S. Albarqouni, and N. Navab, "Steganomaly: Inhibiting cyclegan steganography for unsupervised anomaly detection in brain mri," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 718–727.

[40] D. Zimmerer, F. Isensee, J. Petersen, S. Kohl, and K. Maier-Hein, "Unsupervised anomaly localization using variational auto-encoders," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2019, pp. 289–297.

[41] K. M. van Hespen, J. J. Zwanenburg, J. W. Dankbaar, M. I. Geerlings, J. Hendrikse, and H. J. Kuijf, "An anomaly detection approach to identify chronic brain infarcts on mri," *Scientific Reports*, vol. 11, no. 1, pp. 1–10, 2021.

[42] C. Han, L. Rundo, K. Murao, T. Noguchi, Y. Shimahara, Z. Á. Milacski, S. Koshino, E. Sala, H. Nakayama, and S. Satoh, "Madgan: unsupervised medical anomaly detection gan using multiple adjacent brain mri slice reconstruction," *BMC bioinformatics*, vol. 22, no. 2, pp. 1–20, 2021.

[43] S. You, K. C. Tezcan, X. Chen, and E. Konukoglu, "Unsupervised lesion detection via image restoration with a normative prior," in *International Conference on Medical Imaging with Deep Learning*. PMLR, 2019, pp. 540–556.

[44] L. Zuo, A. Carass, S. Han, and J. L. Prince, "Automatic outlier detection using hidden markov model for cerebellar lobule segmentation," in *Medical Imaging 2018: Biomedical Applications in Molecular, Structural, and Functional Imaging*, vol. 10578. International Society for Optics and Photonics, 2018, p. 105780D.

[45] S. G. Popescu, D. J. Sharp, J. H. Cole, K. Kamnitsas, and B. Glocker, "Distributional gaussian process layers for outlier detection in image segmentation," in *International Conference on Information Processing in Medical Imaging*. Springer, 2021, pp. 415–427.

[46] C. Bowles, C. Qin, R. Guerrero, R. Gunn, A. Hammers, D. A. Dickie, M. V. Hernández, J. Wardlaw, and D. Rueckert, "Brain lesion segmentation through image synthesis and outlier detection," *NeuroImage: Clinical*, vol. 16, pp. 643–658, 2017.

[47] J. Zhang, Y. Xie, G. Pang, Z. Liao, J. Verjans, W. Li, Z. Sun, J. He, Y. Li, C. Shen *et al.*, "Viral pneumonia screening on chest x-rays using confidence-aware anomaly detection," *IEEE transactions on medical imaging*, vol. 40, no. 3, pp. 879–890, 2020.

[48] K. Ouardini, H. Yang, B. Unnikrishnan, M. Romain, C. Garcin, H. Zenati, J. P. Campbell, M. F. Chiang, J. Kalpathy-Cramer, V. Chandrasekhar *et al.*, "Towards practical unsupervised anomaly detection on retinal images," in *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data*. Springer, 2019, pp. 225–234.

[49] G. Quellec, M. Lamard, M. Cozic, G. Coatrieux, and G. Cazuguel, "Multiple-instance learning for anomaly detection in digital mammography," *Ieee transactions on medical imaging*, vol. 35, no. 7, pp. 1604–1614, 2016.

[50] C.-T. Kuo and I. Davidson, "A framework for outlier description using constraint programming," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[51] C. Bowles, C. Qin, C. Ledig, R. Guerrero, R. Gunn, A. Hammers, E. Sakka, D. A. Dickie, M. V. Hernández, N. Royle *et al.*, "Pseudo-healthy image synthesis for white matter lesion segmentation," in *International Workshop on Simulation and Synthesis in Medical Imaging*. Springer, 2016, pp. 87–96.

[52] C. Baur, S. Denner, B. Wiestler, N. Navab, and S. Albarqouni, "Autoencoders for unsupervised anomaly segmentation in brain mr images: a comparative study," *Medical Image Analysis*, p. 101952, 2021.

[53] M. Kim, J. Yun, Y. Cho, K. Shin, R. Jang, H.-j. Bae, and N. Kim, "Deep learning in medical imaging," *Neurospine*, vol. 16, no. 4, p. 657, 2019.

Deep Learning Frameworks Applied For Audio-Visual Scene Classification

Lam Pham, Alexander Schindler, Mina Schütz, Jasmin Lampert, Sven Schlarb and Ross King
Austrian Institute of Technology, 1210 Vienna, Austria
{lam.pham, alexander.schindler, mina.schuetz, jasmin.lampert, sven.schlarb, ross.king}@ait.ac.at

***Abstract*—In this paper, we present deep learning frameworks for audio-visual scene classification (SC) and indicate how individual visual, audio features as well as their combination affect SC performance. Our extensive experiments are conducted on DCASE 2021 (IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events) Task 1B Development and Evaluation datasets. Our results on Development dataset achieve the best classification accuracy of 82.2%, 91.1%, and 93.9% with audio input only, visual input only, and both audio-visual input, respectively. The highest classification accuracy of 93.9%, obtained from an ensemble of audio-based and visual-based frameworks, shows an improvement of 16.5% compared with DCASE 2021 baseline. Our best results on Evaluation dataset is 91.5%, outperforming DCASE baseline of 77.1%**

***Index Terms*—Audio-visual scene, pre-trained model, ImageNet, AudioSet, deep learning framework.**

I. Introduction

Analysing both audio and visual (or image) information from videos has opened a variety of real-life applications such as detecting the sources of sound in videos [1], lip-reading by using audio-visual alignment [2], or source separation [3]. Joined audio-visual analysis shows to be effective compared to the visual only data proven in tasks of video classification [4], multi-view face recognition [5], emotion recognition [6], or video recognition [7]. Although a number of audio-visual datasets exist, they mainly focus on human for specific tasks such as detecting human activity [8], action recognition [9], classifying sport types [10], [11], or emotion detection [6]. Recently, the DCASE community [12] has released an audio-visual dataset proposed for DCASE 2021 Task 1B challenge which aims to classify ten different scene contexts [13]. We therefore evaluate this dataset by leveraging deep learning techniques, then present main contributions following: (1) We evaluate various deep learning frameworks for audio-visual scene classification (SC), indicate individual roles of visual and audio features as well as their combination within SC task; (2) We then propose an ensemble of audio-based and visual-based frameworks, which outperforms the DCASE baseline and are competitive with the state-of-the-art systems; and (3) We evaluate whether the ensemble proposed is effective for detecting scene contexts early.

The paper is organized as follows: Section 2 presents deep learning frameworks proposed for separate audio and visual data input. Section 3 introduces the evaluation setup where the proposed experimental setting, metric, and implementation of deep learning frameworks are presented. Next, Section 4 presents and analyses the experimental results. Finally, Section 5 presents conclusion and future work.

TABLE I
The VGG14 network architecture used for audio-spectrogram based frameworks (input patch of 128×128×6)

Network architecture	Output
BN - Conv [3×3]@64 - ReLU - BN - Dr (25%)	128×128×64
BN - Conv [3×3]@64 - ReLU - BN - AP - Dr (25%)	64×64×128
BN - Conv [3×3]@128 - ReLU - BN - Dr (30%)	64×64×128
BN - Conv [3×3]@128 - ReLU - BN - AP - Dr (30%)	32×32×128
BN - Conv [3×3]@256 - ReLU - BN - Dr (35%)	32×32×256
BN - Conv [3×3]@256 - ReLU - BN - Dr (35%)	32×32×256
BN - Conv [3×3]@256 - ReLU - BN - Dr (35%)	32×32×256
BN - Conv [3×3]@256 - ReLU - BN - AP - Dr (35%)	16×16×256
BN - Conv [3×3]@512 - ReLU - BN - Dr (35%)	16×16×512
BN - Conv [3×3]@512 - ReLU - BN - Dr (35%)	16×16×512
BN - Conv [3×3]@512 - ReLU - BN - Dr (35%)	16×16×512
BN - Conv [3×3]@512 - ReLU - BN - GAP - Dr (35%)	512
FC - ReLU - Dr (40%)	1024
FC - Softmax	$C = 10$

II. Deep learning frameworks proposed

As we aim to evaluate individual roles of audio and visual features within SC task, deep learning frameworks using either audio or visual input are presented in separate sections.

A. *Audio-based deep learning frameworks*

In audio-based deep learning frameworks proposed, audio recordings are firstly transformed into spectrograms where both temporal and frequency features are presented, referred to as front-end low-level feature extraction. As using an ensemble of either different spectrogram inputs [14]–[18] or different deep neural networks [18]–[20] has been a rule of thumb to improve audio-based SC performance, we therefore propose two approaches for back-end classification, referred to as audio-spectrogram and audio-embedding frameworks.

The audio-spectrogram approach uses three spectrogram transformation methods: Mel filter (MEL) [21], Gammatone filter (GAM) [22], and Constant Q Transform (CQT) [21]. To make sure the three types of spectrograms have the same dimensions, the same setting parameters are used with the filter number, window size and hop size set to 128, 80 ms, 14 ms, respectively. As we have two channels for each audio recording and apply deltas, delta-deltas on individual spectrogram, we finally generate spectrograms of 128×704×6. These spectrograms are then split into ten 50%-overlapping patches of 128 × 128 × 6, each which represents for a 1-second audio segment. To enforce back-end classifiers, mixup data augmentation [23], [24] is applied on these patches of spectrogram before feeding them into a VGGish network for classification as shown in Table I. As shown in Table I, the VGGish network architecture contains sub-blocks which perform

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_6

TABLE II
THE PRE-TRAINED MODELS IN [29] PROPOSED FOR EXTRACTING AUDIO EMBEDDINGS

Pre-trained models	Embedding dimension
1/ CNN14	2048
2/ MobileNetV1	1024
3/ Res1dNet30	2048
4/ Resnet38	2048
5/ Wavegram	2048

TABLE III
THE MLP-BASED NETWORK ARCHITECTURE PROPOSED FOR CLASSIFYING AUDIO/VISUAL EMBEDDINGS

Network architecture	Output
FC - ReLU - Dr (40%)	8192
FC - ReLU - Dr (40%)	8192
FC - ReLU - Dr (40%)	1024
FC - Softmax	$C = 10$

convolution (Conv), batch normalization (BN) [25], rectified linear units (ReLU) [26], average pooling (AV), global average pooling (GAP), dropout (Dr) [27], fully-connected (FC) and Softmax layers. The dimension of Softmax layer is set to $C = 10$ which corresponds to the number of scene context classified. In total, we have 12 convolutional layers and 2 fully-connected layers containing trainable parameters that makes the proposed network architecture like VGG14 [28]. We refer to three audio-spectrogram based frameworks proposed as *audio-CQT-Vgg14, audio-GAM-Vgg14*, and *audio-MEL-Vgg14*, respectively.

In the audio-embedding approach, only the Mel filter is used for generating the MEL spectrogram. the Mel spectrograms are fed into pre-trained models proposed in [29] for extracting audio embedding (i.e. the audio embedding, likely vector, is the output of the global pooling layer in pre-trained models proposed in [29]). In this paper, we select five pre-trained models, which achieved high performance in [29], as shown in Table II, for evaluating the audio-embedding approach. As these five pre-trained models are trained on AudioSet [30], a large-scale audio dataset provided by Google for the task of acoustic event detection (AED), using audio embeddings extracted from these models aims to evaluate whether information of sound events detected and condensed in audio embeddings may be effective for SC task. Finally, the audio embeddings are fed into a MLP-based network architecture, as shown in Table III, for classifying into 10 different scene categories. We refer to five audio-embedding based frameworks proposed as *audio-emb-CNN14, audio-emb-MobileNetV1, audio-emb-Res1dNet30, audio-emb-Resnet38,* and *audio-emb-Wavegram*, respectively.

In both approaches, the final classification accuracy is obtained by applying late fusion of individual frameworks (i.e. an ensemble of three predicted probabilities from *audio-CQT-Vgg14, audio-GAM-Vgg14, audio-MEL-Vgg14*, or an ensemble of five predicted probabilities from *audio-emb-CNN14, audio-emb-MobileNetV1, audio-emb-Res1dNet30, audio-emb-Resnet38, audio-emb-Wavegram*).

TABLE IV
THE NETWORK ARCHITECTURES [31] PROPOSED FOR DIRECTLY TRAINING IMAGE FRAMES OR EXTRACTING IMAGE EMBEDDINGS

Network architectures	Size of image inputs	Embedding dimension
1/ Xception	299×299×3	2048
2/ Vgg19	224×224×3	4096
3/ Resnet50	224×224×3	2048
4/ InceptionV3	299×299×3	2048
5/ MobileNetV2	224×224×3	1280
6/ DenseNet121	299×299×3	1024
7/ NASNetLarge	331×331×3	4032

B. Visual-based deep learning frameworks

Similar to the audio-based frameworks mentioned above, we also propose two approaches for analysing how visual features affect the SC performance: A visual-image approach where classifying process is directly conducted on the image frame inputs, and a visual-embedding approach where the classification is conducted on image embeddings extracted from pre-trained models. In both approaches proposed, we use the same network architectures from Keras application library [31], which are considered as benchmarks for evaluating ImageNet dataset [32] as shown in Table IV. In order to directly train image frame inputs with the network architectures in Table IV, we reduce the C dimensions of the final fully connected layer ($C = 1000$ that equals to the number of object detection defined in ImageNet dataset) to $C = 10$ that matches the number of scene categories classified. The visual-image frameworks proposed are referred to as *visual-image-Xception, visual-image-Vgg19, visual-image-Resnet50, visual-image-InceptionV3, visual-image-MobileNetV2, visual-image-DenseNet121*, and *visual-image-NASNetLarge*, respectively.

Regarding the visual-embedding approach, the network architectures mentioned in Table IV are trained with the ImageNet dataset [32]. Then, image frames of the scene dataset are fed into these pre-trained models to extract image embeddings (i.e. the image embedding, likely vector, is the output of the second-to-last fully connected layer of pre-trained models). Finally, the extracted image embeddings are fed into a MLP-based network architecture as shown in Table III for classifying into ten scene categories (Note that we use the same MLP-based network architecture for classifying audio or image embeddings). The visual-embedding frameworks proposed are referred to as *visual-emb-Xception, visual-emb-Vgg19, visual-emb-Resnet50, visual-emb-InceptionV3, visual-emb-MobileNetV2, visual-emb-DenseNet121,* and *visual-emb-NASNetLarge*, respectively.

Similar to the audio-based approaches, the final classification accuracy of visual-based frameworks is obtained by applying late fusion of individual frameworks (i.e. an ensemble of seven predicted probabilities from seven visual-image based frameworks, or an ensemble of seven predicted probabilities from seven visual-embedding based frameworks).

III. EVALUATION SETTING

A. TAU Urban Audio-Visual Scenes 2021 dataset [13] (Development and Evaluation datasets)

The **Development dataset** is referred to as DCASE Task 1B Development, which was proposed for DCASE 2021 challenge [12]. The dataset is slightly unbalanced and contains both acoustic and visual information, recorded from 12 large European cities: Amsterdam, Barcelona, Helsinki, Lisbon,

TABLE V
The number of 10-second audio-visual scene recordings corresponding to each scene categories in the Train. and Eval. subsets separated from the DCASE 2021 Task 1B Development dataset, and Evaluation dataset [13].

Category	Train.	Eval.	Evaluation
Airport	697	281	-
Bus	806	327	-
Metro	893	386	-
Metro Station	893	386	-
Park	1006	386	-
Public square	982	387	-
Shopping mall	841	387	-
Street pedestrian	968	421	-
Street traffic	985	402	-
Tram	763	308	-
Total	8646 (≈24 hours)	3645 (≈10 hours)	7200 (20 hours)

London, Lyon, Madrid, Milan, Prague, Paris, Stockholm, and Vienna. It consists of 10 scene classes: airport, shopping mall (indoor), metro station (underground), pedestrian street, public square, street (traffic), travelling by tram, bus and metro (underground), and urban park, which can be categorised into three meta-class of indoor, outdoor, and transportation. To evaluate, we follow the DCASE 2021 Task 1B challenge [12], separate this dataset into training (Train.) subset for the training process and evaluation (Eval.) subset for the evaluating process as shown in Table V.

The DCASE 2021 Task 1B challenge also releases the **Evaluation dataset** without labels, which is used to evaluate the submitted systems. The total number of 10-second segments is 7200 (20 hours). In this paper, our results on both Eva. subset and Evaluation dataset (accuracy scoring for Evaluation dataset is conducted by DCASE 2021 task 1A challenge as labels is not released) are reported and compared with the state-of-the-art systems.

B. *Deep learning framework implementation*

Extract audio/visual embeddings from pre-trained models: Since the pre-trained models, which are used for extracting audio embeddings from [29], are built on Pytorch framework, the process of extracting embedding from these models is also implemented with Pytorch framework. Meanwhile, we use the Tensorflow framework for extracting visual embeddings as the pre-trained models are built with the Keras library [31] using back-end Tensorflow.

Classification models for audio/visual data: We use Tensorflow framework to build all classification models in this papers (i.e. Vgg14 and MLP-base network architectures mentioned in Table I and Table III, respectively). As we apply mixup data augmentation [23], [24] for both high-level feature of audio/visual embeddings and low-level feature of audio spectrograms/image frames to enforce back-end classifiers, the labels of the mixup data input are no longer one-hot. We therefore train back-end classifiers with Kullback-Leibler (KL) divergence loss [33] rather than the standard cross-entropy loss over all N mixup training samples:

$$LOSS_{KL}(\Theta) = \sum_{n=1}^{N} \mathbf{y}_n \log\left(\frac{\mathbf{y}_n}{\hat{\mathbf{y}}_n}\right) + \frac{\lambda}{2}||\Theta||_2^2, \qquad (1)$$

where Θ denotes the trainable network parameters and λ denotes the ℓ_2-norm regularization coefficient. $\mathbf{y_c}$ and $\hat{\mathbf{y}}_\mathbf{c}$ denote the ground-truth and the network output, respectively. The training is carried out for 100 epochs using Adam [34] for optimization.

C. *Metric for evaluation*

Regarding the evaluation metric used in this paper, we follow DCASE 2021 Task 1B challenge. Let us consider C as the number of audio/visual test samples which are correctly classified, and the total number of audio/visual test samples is T, the classification accuracy (Acc. (%)) is the % ratio of C to T.

D. *Late fusion strategy for multiple predicted probabilities*

As back-end classifiers work on patches of spectrograms or image frames, the predicted probability of an entire spectrogram or all image frames of a video recording is computed by averaging of all images or patches' predicted probabilities. Let us consider $\mathbf{P^n} = (\mathbf{p_1^n}, \mathbf{p_2^n}, ..., \mathbf{p_C^n})$, with C being the category number and the n^{th} out of N image frames or patches of spectrogram fed into a learning model, as the probability of a test instance, then the average classification probability is denoted as $\bar{\mathbf{p}} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_C)$ where,

$$\bar{p}_c = \frac{1}{N}\sum_{n=1}^{N} p_c^n \quad for \quad 1 \leq n \leq N \qquad (2)$$

and the predicted label $\hat{y}$ for an entire spectrogram or all image frames evaluated is determined as:

$$\hat{y} = argmax(\bar{p}_1, \bar{p}_2, ..., \bar{p}_C) \qquad (3)$$

To evaluate ensembles of multiple predicted probabilities obtained from different frameworks, we proposed three late fusion schemes, namely MEAN, PROD, and MAX fusions. In particular, we conduct experiments over individual frameworks, thus obtain the predicted probability of each framework as $\bar{\mathbf{p}}_\mathbf{s} = (\bar{p}_{s1}, \bar{p}_{s2}, ..., \bar{p}_{sC})$ where C is the category number and the s^{th} out of S framework evaluated. Next, the predicted probability after late MEAN fusion $\mathbf{p_{f-mean}} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_C)$ is obtained by:

$$\bar{p_c} = \frac{1}{S}\sum_{s=1}^{S} \bar{p}_{sc} \quad for \quad 1 \leq s \leq S \qquad (4)$$

The PROD strategy $\mathbf{p_{f-prod}} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_C)$ is obtained by,

$$\bar{p_c} = \frac{1}{S}\prod_{s=1}^{S} \bar{p}_{sc} \quad for \quad 1 \leq s \leq S \qquad (5)$$

and the MAX strategy $\mathbf{p_{f-max}} = (\bar{p}_1, \bar{p}_2, ..., \bar{p}_C)$ is obtained by,

$$\bar{p_c} = max(\bar{p}_{1c}, \bar{p}_{2c}, ..., \bar{p}_{Sc}) \qquad (6)$$

Finally, the predicted label $\hat{y}$ is determined by (3):

TABLE VI
PERFORMANCE COMPARISON OF AUDIO-BASED FRAMEWORKS

Audio-spectrogram based models	**Acc.**	**Audio-embedding based models**	**Acc.**
audio-CQT-Vgg14	68.3	audio-emb-CNN14	64.4
audio-GAM-Vgg14	69.6	audio-emb-MobileNetV1	57.8
audio-MEL-Vgg14	72.2	audio-emb-Res1dNet30	58.0
		audio-emb-Resnet38	62.7
		audio-emb-Wavegram	63.4
MAX Fusion	78.0	MAX Fusion	64.9
MEAN Fusion	79.7	MEAN Fusion	**69.6**
PROD Fusion	**80.4**	PROD Fusion	68.4

TABLE VII
PERFORMANCE COMPARISON OF VISUAL-BASED FRAMEWORKS

Visual-image based models	**Acc.**	**Visual-embedding based models**	**Acc.**
visual-image-Xception	85.9	visual-emb-Xception	80.3
visual-image-Vgg19	83.8	visual-emb-Vgg19	80.8
visual-image-Resnet50	86.3	visual-emb-Resnet50	82.0
visual-image-InceptionV3	88.9	visual-emb-InceptionV3	83.4
visual-image-MobileNetV2	84.4	visual-emb-MobileNetV2	80.2
visual-image-DenseNet121	87.8	visual-emb-DenseNet121	83.5
visual-image-NASNetLarge	86.9	visual-emb-NASNetLarge	81.5
MAX Fusion	90.2	MAX Fusion	**86.5**
MEAN Fusion	90.5	MEAN Fusion	81.8
PROD Fusion	**91.1**	PROD Fusion	84.3

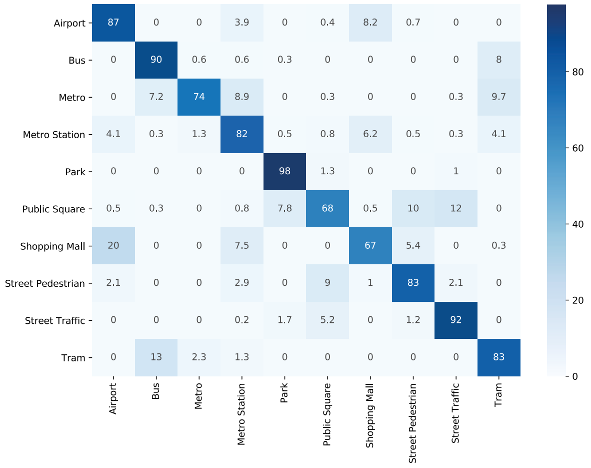


Fig. 1. Confusion matrix result (Acc. %) obtained by PROD fusion of *audio-CQT-Vgg14, audio-GAM-Vgg14, audio-MEL-Vgg14,* and *audio-emb-CNN14*

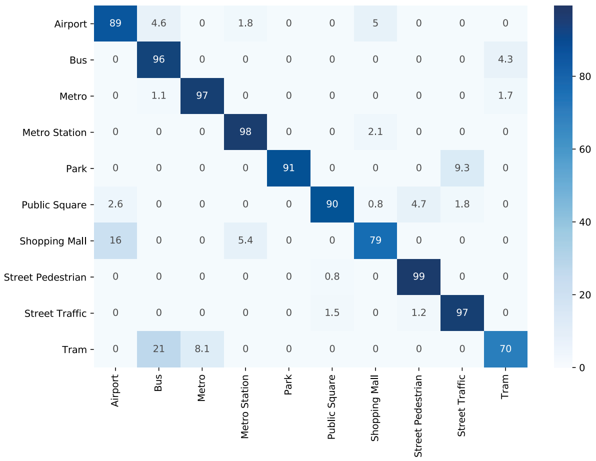


Fig. 2. Confusion matrix result (Acc. %) obtained by PROD fusion of seven visual-image based frameworks

IV. Experimental Results And Discussion

A. Analysis of audio-based deep learning frameworks for scene classification

As Table VI shows the accuracy results obtained from audio-based deep learning frameworks, we can see that all late fusion methods help to improve the performance significantly regarding both audio-spectrogram and audio-embedding approaches, achieving the highest score of 80.4% from PROD fusion of three audio-spectrogram based frameworks and 69.6% from MEAN fusion of five audio-embedding based frameworks. Compare the performance between two audio-based approaches proposed, it can be seen that directly training spectrogram inputs is more effective, achieving 68.3%, 69.6%, and 72.2% from CQT, GAM, and MEL spectrogram respectively, which outperform all results obtained from audio-emmbedding based frameworks. We further conduct PROD fusion of predicted probabilities obtained from three audio-spectrogram based frameworks (*audio-CQT-Vgg14, audio-GAM-Vgg14, audio-MEL-Vgg14*) and the *audio-emb-CNN14* framework (the best framework in the audio-embedding based approach), achieving the classification accuracy of 82.2% with the confusion matrix shown in Fig. 1 and improving the DCASE baseline by 17.1% (Note that only audio data input is used for these frameworks and DCASE baseline). This proves that although the audio-embedding based approach gains low performance rather than the audio-spectrogram based approach, audio embeddings extracted from AudioSet dataset for AED task contain distinct features which is beneficial for SC task.

B. Analysis of visual-based deep learning frameworks for scene classification

As obtained results are shown in Table VII, we can see that the visual-image based frameworks, which directly train image frame inputs, outperform visual-embedding based frameworks. While all late fusion methods over visual-image based frameworks help to improve the performance, only MAX fusion of image-embedding based frameworks shows to be effective. The PROD fusion of seven visual-image based frameworks achieves the best accuracy of 91.1%, improving DCASE baseline by 13.7% (Note that these frameworks and DCASE baseline only use visual data input). Comparing the performance between audio-based and visual-based approaches, the PROD fusion of seven visual-image based frameworks (91.1%) outperforms the best result of 82.2% from PROD fusion of *audio-CQT-Vgg14, audio-GAM-Vgg14, audio-MEL-Vgg14,* and *audio-emb-CNN14* mentioned in Section IV-A. Further comparing the two confusion matrixes obtained from these two PROD fusions, as shown in Fig. 1 and Fig. 2, we can see that PROD fusion of seven visual-image based frameworks outperforms over almost scene categories except to 'Park' and 'Tram'. As a result, we can conclude that visual data input

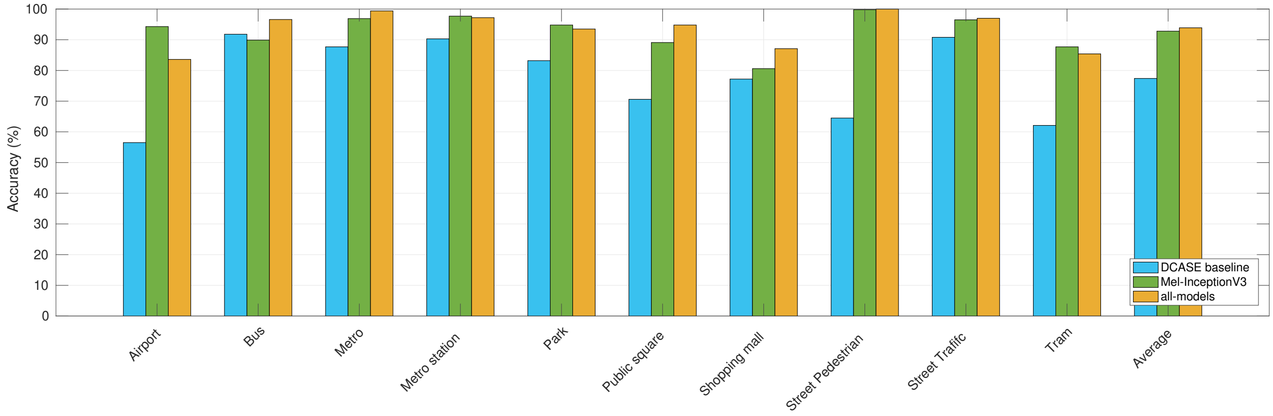


Fig. 3. Performance comparison (Acc.%) of DCASE baseline, *MEL-InceptionV3* and *all-models* across all scene categories

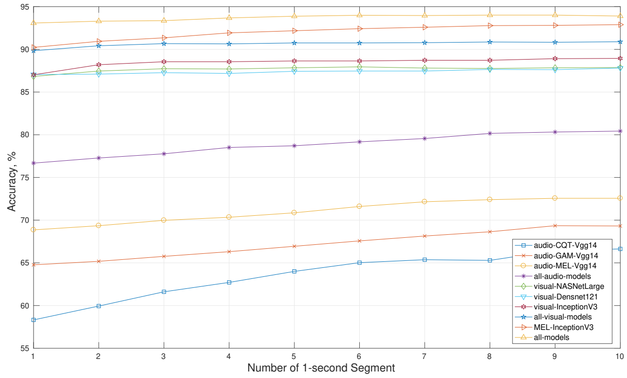


Fig. 4. Performance of individual audio-spectrogram based frameworks (*audio-CQT-Vgg14, audio-GAM-Vgg14, audio-MEL-Vgg14*), PROD fusion of three audio-spectrogram based frameworks (*all-audio-models*), individual visual-image based frameworks (*visual-image-NASNetLarnge, visual-image-Densnet121, visual-image-InceptionV3*), PROD fusion of three visual-image based frameworks (*all-visual-model*), PROD fusion of audio-based and visual based frameworks (*MEL-InceptionV3, all-models*) with the increasing number of 1-second input segments

contains more information for scene classification rather than audio data input.

C. Combine both visual and audio features for scene classification

As individual analysis of either audio or visual features within scene context classification is shown in Section IV-A and IV-B respectively, we can see that directly training and classifying audio/visual data input is more effective, rather than audio/image-embedding based approaches. We then evaluate a combination of audio and visual features by proposing two PROD fusions: (1) three audio-spectrogram based frameworks (*audio-CQT-Vgg14, audio-GAM-Vgg14, audio-MEL-Vgg14*) and top-3 visual-image frameworks (*visual-image-DenseNet121, visual-image-InceptionV3, visual-image-NASNetLarge*) referred to as *all-models*, and (2) one audio-spectroram based framework (*audio-MEL-Vgg14*) and one visual-image based framework (*visual-image-InceptionV3*) referred to as *MEL-InceptionV3*. As results shown in Fig. 3, *all-models* helps to achieve the highest accuracy classification score of 93.9%, improving DCASE baseline by 16.5% and showing improvement on all scene categories. Although *MEL-InceptionV3* only fuses two frameworks, it achieves 92.8%, showing competitive to *all-models* fusing 6 frameworks. Notably, misclassification cases mainly occur among high cross-correlated categories in meta-class such as indoor (Airport, Metro station, and Shopping mall), outdoor (Park, Public square, Street pedestrian, and Street Traffic), and transportation (Bus, Metro, and Tram). If we aim to classify into three meta-classes (indoor, outdoor, and transportation), *all-models* helps to achieve a classification accuracy of 99.3%.

D. Early detecting scene context

We further evaluate whether deep learning frameworks can help to detect scene context early. To this end, we evaluate 10 different frameworks: (1-2-3) 3 individual audio-spectrogram based frameworks (*audio-CQT-Vgg14, audio-GAM-Vgg14, audio-MEL-Vgg14*), (4) PROD fusion of these three audio-spectrogram frameworks referred to as *all-audio-models*, (5-6-7) 3 visual-image based frameworks (*visiual-image-NASNetLarge, visual-image-Densnet121, visual-image-InceptionV3*), (8) PROD fusion of these three visual-image based frameworks referred to as *all-visual-models*, (9) *MEL-InceptionV3*, and (10) *all-models*. As the results are shown in Fig. 4, while performance of audio-based frameworks is improved by time, visual-based frameworks are stable. As a result, when we combine audio and visual features, which are evaluated in *MEL-InceptionV3* and *all-models*, the performance is improved by time and stable after 6 seconds. Notably, accuracy scores of both *MEL-InceptionV3* and *all-models* are larger than 90.0% at the first second, which is potentially for real-life applications integrating the function of early detecting scene context.

E. Compare with the state-of-the-art systems

Compare with the state-of-the-art systems, our result on Eva. subset achieves 93.9%, occupying the top-5 team ranking with respect to accuracy performance. Similarly, our accuracy result on Evaluation dataset is 91.5%, also achieving the top-5 team ranking. Furthermore, the low gap of accuracy performance between Eval. Subset (93.9%) and Evaluation dataset (91.5%) proves our proposed framework robust and general.

V. Conclusion

We conducted extensive experiments and explored various deep learning based frameworks for classifying 10 categories of urban scenes. Our method, which uses an ensemble of audio-based and visual-based frameworks, achieves the best classification accuracy of 93.9% on DCASE Task 1B Development dataset and 91.5% on DCASE task 1B Evaluation dataset. The obtained results outperform DCASE baseline, improving by 17.1% with only audio data input, 26.2% with only visual data input, and 16.5% with both audio-visual data on Development dataset, and improving by 14.3% with both audio-visual data for Evaluation dataset. In further work, we will evaluate whether an end-to-end system using joint learning of audio-visual data input may help to improve the performance.

Acknowledgement

The AMMONIS project is funded by the FORTE program of the Austrian Research Promotion Agency (FFG) and the Federal Ministry of Agriculture, Regions and Tourism (BMLRT) under grant no. 879705.

References

[1] R. Arandjelović and A. Zisserman, “Objects that sound,” in *ECCV*, 2018.

[2] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, “Lip reading sentences in the wild,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3444–3453.

[3] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. H. McDermott, and A. Torralba, “The sound of pixels,” *ArXiv*, vol. abs/1804.03160, 2018.

[4] N. Takahashi, M. Gygli, and L. V. Gool, “Aenet: Learning deep audio features for video analysis,” *IEEE Transactions on Multimedia*, vol. 20, pp. 513–524, 2018.

[5] C. Sanderson and B. Lovell, “Multi-region probabilistic histograms for robust and scalable identity inference,” in *Proc. International conference on biometrics (ICB)*, 2009.

[6] S. Zhang, S. Zhang, T. Huang, W. Gao, and Q. Tian, “Learning affective features with a hybrid deep model for audio–visual emotion recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, pp. 3030–3043, 2018.

[7] R. Gao, T.-H. Oh, K. Grauman, and L. Torresani, “Listen to look: Action recognition by previewing audio,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 454–10 464.

[8] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, “Activitynet: A large-scale video benchmark for human activity understanding,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 961–970.

[9] K. Soomro, A. Zamir, and M. Shah, “Ucf101: A dataset of 101 human actions classes from videos in the wild,” *ArXiv*, vol. abs/1212.0402, 2012.

[10] R. Gade, M. Abou-Zleikha, M. G. Christensen, and T. Moeslund, “Audio-visual classification of sports types,” in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015, pp. 768–773.

[11] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.

[12] Detection and Classification of Acoustic Scenes and Events Community, *DCASE 2021 challenges*, http://dcase.community/challenge2021.

[13] S. Wang, A. Mesaros, T. Heittola, and T. Virtanen, “A curated dataset of urban scenes for audio-visual scene analysis,” *arXiv preprint arXiv:2011.00030*, 2020.

[14] L. Pham, I. Mcloughlin, H. Phan, R. Palaniappan, and A. Mertins, “Deep feature embedding and hierarchical classification for audio scene classification,” in *International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–7.

[15] L. Pham, H. Phan, T. Nguyen, R. Palaniappan, A. Mertins, and I. Mcloughlin, “Robust acoustic scene classification using a multi-spectrogram encoder-decoder framework,” *Digital Signal Processing*, vol. 110, p. 102943, 2021.

[16] L. Pham, I. McLoughlin, H. Phan, R. Palaniappan, and Y. Lang, “Bag-of-features models based on C-DNN network for acoustic scene classification,” in *Proc. International Conference on Audio Forensics (AES)*, 2019.

[17] L. Pham, I. Mcloughlin, H. Phan, and R. Palaniappan, “A robust framework for acoustic scene classification,” in *Proc. International Speech Communication Association (INTERSPEECH)*, 2019, pp. 3634–3638.

[18] H. Phan, H. Le Nguyen, O. Y. Chén, L. Pham, P. Koch, I. McLoughlin, and A. Mertins, “Multi-view audio and music classification,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 611–615.

[19] D. Ngo, H. Hoang, A. Nguyen, T. Ly, and L. Pham, “Sound context classification basing on join learning model and multi-spectrogram features,” *ArXiv*, vol. abs/2005.12779, 2020.

[20] T. Nguyen and F. Pernkopf, “Acoustic scene classification using a convolutional neural network ensemble and nearest neighbor filters,” in *Proc. DCASE*, 2018, pp. 34–38.

[21] B. McFee, R. Colin, L. Dawen, D. Ellis, M. Matt, B. Eric, and N. Oriol, “librosa: Audio and music signal analysis in python,” in *Proceedings of The 14th Python in Science Conference*, 2015, pp. 18–25.

[22] D. P. W. . Ellis, “Gammatone-like spectrogram,” 2009. [Online]. Available: http://www.ee.columbia.edu/ dpwe/resources/matlab/ gammatonegram

[23] K. Xu, D. Feng, H. Mi, B. Zhu, D. Wang, L. Zhang, H. Cai, and S. Liu, “Mixup-based acoustic scene classification using multi-channel convolutional neural network,” in *Pacific Rim Conference on Multimedia*, 2018, pp. 14–23.

[24] Y. Tokozume, Y. Ushiku, and T. Harada, “Learning from between-class examples for deep sound recognition,” in *International Conference on Learning Representations (ICLR)*, 2018.

[25] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 448–456.

[26] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *International Conference on Machine Learning (ICML)*, 2010.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations (ICLR)*, 2015.

[29] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, “Panns: Large-scale pretrained audio neural networks for audio pattern recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[30] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, “Audio set: An ontology and human-labeled dataset for audio events,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[31] F. Chollet *et al.*, “Keras,” https://keras.io, 2015.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[33] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[34] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6980, 2015.

Security and Data Integrity in Machine Learning

Toward Applying the IEC 62443 in the UAS for Secure Civil Applications

Abdelkader Magdy Shaaban*, Oliver Jung* and Miguel Angel Fas Millan†
*AIT Austrian Institute of Technology, 1210 Vienna, Austria
†German Aerospace Center (DLR), 38108 Brunswick, Germany
{abdelkader.shaaban, Oliver.Jung}@ait.ac.at,
miguelangel.fasmillan@dlr.de

***Abstract*—The growing demand for drones in civil applications is usually satisfied with commercial off-the-shelf devices. These can always be adapted to meet the final user's needs, but they could not satisfy critical aspects such as performance, efficiency, or security. Cybersecurity is one of the critical issues in Unmanned Aircraft Systems (UAS), where cyberattacks on this system could lead to multiple negative consequences. We address the cybersecurity issue in this work by introducing a set of strategic actions to define a complete development process of building secure Unmanned Aerial Vehicle (UAV) applications. We introduce the first steps toward implementing IEC 62443 security standard in UAS. We create comprehensive threats, components and critical assets catalogues for UAVs. Then, we employ the ThreatGet tool to automatically identify and determine relevant threats and estimate risk severities associated with a UAS case study. ThreatGet's findings are then used to outline a mapping procedure between threats and security requirements. This strategy aims to identify a set of security requirements to address potential threats and protect critical assets in UAS.**

***Index Terms*—UAV, Potential Threats, Security Requirements, Risk Management, Cybersecurity.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) have a growing presence in civil applications domains. They are not only used in agriculture or for infrastructure maintenance and surveillance but are also considered as a means of transport, including urban areas. The Single European Sky ATM Research (SESAR) project, in charge of the modernization of the European airspace, expects a fleet of 400 000 drones to be used for commercial and government missions in 2050 [1]. For managing the access to the airspace for large numbers of drones efficiently, SESAR promoted the development of the U-space concept of operations (ConOps), which sets the baselines for a harmonized, safe, efficient, respectful and secure integration of the drones in the airspace. Regarding security, the ConOps ([2], 4.6 Cyber security of U-space) stresses the need to mitigate the risks concerning the following five aspects. The most important aspect is the **integrity**. A second aspect is the **availability**; a typical measure to ensure the continuity of the service is redundancy. **Confidentiality** of information stored or in transfer should be maintained, e.g only authorized users or services should be able to gain access. **Security awareness** is another aspect that should be boosted, operators as well as drone pilots should undergo encouraging cyber security trainings. The last aspect is **enforcement**, which includes monitoring operations to verify instruction compliance or the possibility of identifying rogue drone that could imply a risk for people or infrastructures. Given this, the goal of the Labyrinth project [1] is to implement and test in real scenarios [3] some of the services and procedures that are part of the U-space [2] ConOps, taking into consideration aspects like the performance of communications.

In order to achieve a specific mission, UAVs need communication channels to communicate to other nodes (e.g., UAVs, roadside base stations, central base stations, etc.). Any single vulnerable point in the system design could lead to multiple ways for attackers to perform malicious activities. Therefore, it is necessary to define applicable security requirements for each system's node to protect the whole system against cyberattacks and address existing security vulnerabilities. However, integrating requirements into system design is considered a challenging process since these requirements could be redundant or unsuitable for addressing identified security issues. There are many existing security standards from related domains that can be used to build secure UAS applications, such as the ISO27000 family [4], Common Criteria [5], and the IEC 62443 family [6]. The IEC 62443 family presents procedures for implementing secure Industrial Automation and Control Systems (IACS).

Therefore, this work introduces the first steps into adopting IEC 62443 security standard in the UAS. We apply IEC 62443-4-2 [7] in order to secure the UAS on the components level. In order to achieve the main target of this work, we propose a threat modelling approach to assist in applying IEC 62443 in an UAS. We define security zones and conduits in the system design and specify the security requirements according to the Foundational Requirements (FRs) defined in IEC 62433. Each security zone and conduit has particular Security Targets (ST) that need to be achieved. These targets are estimated according to the impact of each zone and conduit from multiple potential threats. Furthermore, to estimate STs and define the existing security vulnerabilities in the system design, we use the ThreatGet threat analysis tool. ThreatGet is a plugin for the Enterprise Architect UML modelling tool

[1] http://labyrinth2020.eu

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_7

developed by AIT - Austrian Institute of Technology [2] [8]. It analyses the security-related vulnerabilities in a system model and estimates the risk severity for each of the identified threats. The tool also classifies threats according to the STRIDE model. The outcomes of ThreatGet help in achieving the main objective of this work by:

1) Estimating the security target for each zone/conduit according to the risk degree of the identified threats.
2) Defining the security property violations to assist in mapping security requirements for addressing existing potential threats based on the mapping between FRs and STRIDE classification.

According to the proposed work's findings, we can estimate ST for each zone/conduit and easily describe a specific set of security requirements that can handle existing security vulnerabilities to meet the actual security goal.

A. Related Work

As a response to the SESAR 2020 RPAS Exploratory Research Call, under Topic 06 Security & cyber-resilience, the SECOPS project (SECurity concept for drone OPerationS) [9] developed a methodology based on the SESAR2020 Security Risk Assessment methodology (SecRAM 2.0) [10] to evaluate the risk of the operations focusing on cyber security threats. SECOPS suggested requirements, mitigations, and security controls for U-space. Security controls deal with medium and high risks; these risks are classified according to a combination of likelihood and impact values. The impact represents the harm that a threat can cause to U-space services regarding confidentiality, integrity, or availability.

However, SECOPS is not the only security assessment method for U-space. The same U-space ConOps includes a MEthoDology for the USpace Safety Assessment (MEDUSA). MEDUSA is a holistic approach, considering the drone and operator perspective, the Unmanned Traffic Management (UTM) service provisioning, and its coordination with the manned Air Traffic Management. The UTM is the implementation of the U-space concept, a system to control the unmanned traffic. In short, it can be seen like an automated version of the air traffic controllers but applied to small drones. This system does not necessarily run in a single server, and its different functions (called services in the ConOps) can be distributed, and performed by different companies. It also makes use of external information, like meteorological reports or terrain elevation maps, to perform its tasks. This system must also be coordinated with human air traffic controllers, especially when manned aviation needs to enter the very low level altitude airspace volumes occupied by the drones. Therefore, the UTM implies an exchange of critical information between different actors.

The drone and operator safety perspective is taken from the Specific Operational Risk Assessment methodology (SORA) [11] outcome, which is applied during the flight planning process to determine the risk of the operation and the possible mitigation mechanisms. The UTM viewpoint is based on EUROCONTROL's Safety Reference Material (SRM) [12]. The purpose of MEDUSA is to identify and develop safety requirements and recommendations for U-space. This identification is made using two approaches; one considering the system in the absence of failures and a second evaluating the risks that the U-space could generate in case of a failure in the system. It starts by determining the acceptable level of safety for each risk during the operation.

Finally, it is worth mentioning the approach of the Federal Aviation Administration's Security Considerations for Operationalization of the UTM Architecture as part of its Mid Atlantic Aviation Partnership (MAAP) UTM Pilot Program Phase 2 (UPP2) [13], which has some common points with SECOPS. In particular, this risk analysis is based on the ISO 27005 Risk Management Standard [14] and the NIST SP800 series of security standards [15]. In it, three domains are defined: trusted parties, including UAS Service Suppliers providing critical services to the UTM network; intermediate trusted parties, like Supplemental Data Service Providers (SDSP), accessing to the network through Virtual Private Networks and authenticated services; and other third parties, which could include other services or consumers of data. Each domain with its associated security controls and processes for authentication and interaction with other parties.

The topic of threat analyses in the UAS has been adopted in multiple types of research. Ref. [16] identifies a new set of potential threats for UAVs, which require careful security considerations in the development process. A set of UAV challenges regarding safety, security,privacy, and liability is discussed in [17]. The authors developed a set of recommendations to address and tackle these challenges. Multiple security threats in the UAS system are discussed and presented in [18], which provides a comprehensive understanding of threats and related mitigation mechanisms. Cybersecurity vulnerabilities could be identified within the strategic, operational, acquisition, and tactical levels. Ref. [19] focused on the tactical level for defining a cybersecurity assessment. Also, [20] presented multiple cyberattacks and relevant countermeasure. The Afarcloud [3] project finally investigated a set of potential threats relevant to the UAS to be considered for building secure smart farming applications [21].

II. Applying IEC 62443 Security Standard in UAS

Cybersecurity plays a vital role in the UAS domain because it protects data and critical units responsible for controlling the UAV's functional safety from various attack scenarios. Accordingly, the safety-security relationship is considered directly proportional any malicious activity against the UAS network could lead to safety hazards against civilians, infrastructure, and other targets. For instance, attackers could compromise transmitted commands; and the UAV might then receive falsified commands. This attack could jeopardize the safety of UAV's operations and cause it to act as a weapon

[2] https://www.ait.ac.at/en/

[3] http://www.afarcloud.eu

by injuring people or damaging infrastructure. Also, some other scenarios could be expected, such as camera hijacking when critical cybersecurity properties are exploited. Therefore, we use ThreatGet for identifying potential threats and relevant security vulnerabilities in the UAS. For this purpose, we built a complete database containing a wide range of potential threats in the UAS domain based on the state-of-the-art [18], [19], [22], [20], [21], [23], [24], and [25].

Then we address the potential threats obtained by ThreatGet by applying IEC 62443 to the UAS-domain to guarantee a satisfied protection level and to avoid any unanticipated adverse outcomes.

We formalize these actions in a set of steps to define a complete development process for secure UAV applications. Figure 1 illustrates a flowchart that represents the proposed strategic actions for building a secure UAV infrastructure.

The figure depicts the primary steps described in this paper for adopting IEC 62443 and selecting the appropriate sets of requirements for the UAS.

A. Assets Identification

As a part of our research in the Labyrinth project, we investigate the most common components (i.e., elements, connectors, and critical assets) that can be used to model UAS examples. An element is defined as a physical or logical item of a system model, whereas connectors define the data flow between many elements in the system design. In addition, it is necessary to identify UAS assets that need more security concerns. An asset means something valuable for the stakeholder, which needs more security measures to protect it from various malicious actions. On the other hand, an asset is a worthwhile target for attackers (i.e., information, signals, configurations, collected images, etc.). Therefore, a comprehensive component catalogue for ThreatGet is created to build multiple UAS application models. A simple example of the UAS model is depicted in Figure 2.

The figure illustrates the data flow among different terminals in a U-space framework. On the right side, the Ground Control Station (GCS) is identified as the primary central unit responsible for controlling UAV flight activities. A set of commands that regulate the UAV's travel directions can be sent from the GCS to the UAV. The GCS can also receive data from the UAV through a wireless connection, such as UAV's telemetry data. Additionally, the UAV can transmit reports to the UTM (on the left side) with information on its flight status. These reports can be sent to the GCS via an Internet connection to keep the GCS up to date on the UAV's mission flight status. The GCS can also receive instructions from the UTM and then translate them into commands to the UAV. The "A" letters in the figure indicate assets, defined as critical items requiring more security concern. Each component of this model is contained inside a security zone referred to as a boundary. ThreatGet specifies the UAV, UTM, and GCS boundaries as "Boundary [USER]" because of the user created for that purpose.

Each asset in the UAS has security properties representing protection against different attack scenarios, such as authentication, authorization, temper protection, encryption, etc. The system architect can specify these properties to provide a set of risk mitigation actions against potential threats. ThreatGet checks the violations of these properties and indicates if there is a security gap that could allow a malicious activity.

B. Identify Security Zones

Identifying security zones is essential in defining a physical or logical segmentation of the system design. These zones consist of a set of system assets that share the corresponding security requirements [6]. According to IEC 62443-4-2 [7], seven FR classes described the security requirements. The FR5 - Restricted Data Flow (RDF) describes constraints of unnecessary data flows to limit the spread of any cyberattacks in the form of a set of zones. In addition, data transmission between zones is accomplished via communication channels; these channels are grouped into small clusters of zones called conduits. Furthermore, in our example, we split the UAS model into a set of zones and conduits according to the FR5. ThreatGet helps in describing these security zones as boundaries (i.e., UTM, UAV, and GCS), as depicted in Figure 2. We create a tagged value "Security Conduit" for each communication channel to build conduits for communication channels between zones. The "Security Conduit" has three values (i.e., UTM-UAV, UTM-GCs, and UAV-GCS) that indicate to which conduits this communication channel belongs. For example, all communication channels between the UAV and GCS shall have the "UAV-GCS" value assigned to the "Security Conduit, " which indicate that all communication channels exist in the same conduit.

In this work, we apply a direct mapping approach between potential security threats and relevant security requirements based on the FRs; these FRs are discussed in Section II-E.

C. Risk Analysis

The risk analysis plays a significant role to identify and determine the exact cybersecurity issues in the UAV system model, leading to different types of potential cyber threats. Therefore, we build a threat database for the UAS based on the related work, which ThreatGet can utilize to identify relevant threats in various UAS case studies. All threats are translated into a formal grammatical structure in order to automate the threat analysis and identify affected units. For example, as described in Figure 2, the UAV receives commands from the GCS through the data flow; attackers could falsify these commands by compromising the integrity of the transferred messages. We can formalize this attack using the ThreatGet's grammar by considering a data flow crossing security zone. List 1 illustrates as a simple snippet of ThreatGet's grammar. The "connector pattern,", is a connector that has a source element (i.e., source filter) and a target element (i.e., target filter). Each source and target element contain a collection of security properties expressed as a combination of tagged values. Each tagged value may have a

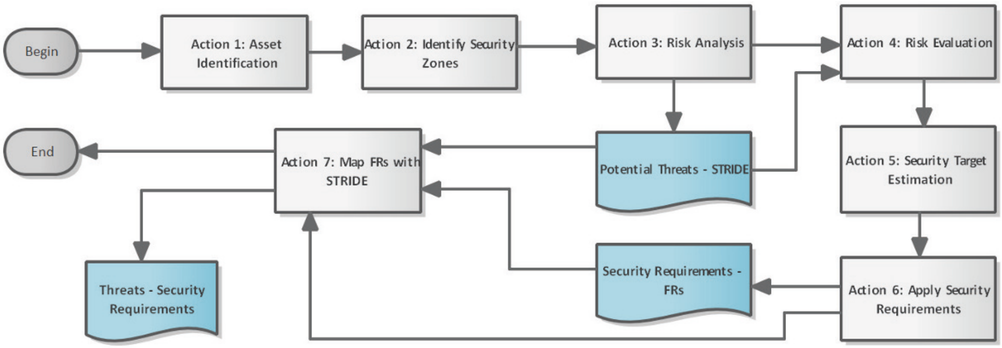

Begin
Action 1: Asset Identification
Action 2: Identify Security Zones
Action 3: Risk Analysis
Action 4: Risk Evaluation
End
Action 7: Map FRs with STRIDE
Potential Threats - STRIDE
Action 5: Security Target Estimation
Threats - Security Requirements
Security Requirements - FRs
Action 6: Apply Security Requirements

Fig. 1: The proposed steps of applying IEC 62443 for the UAS

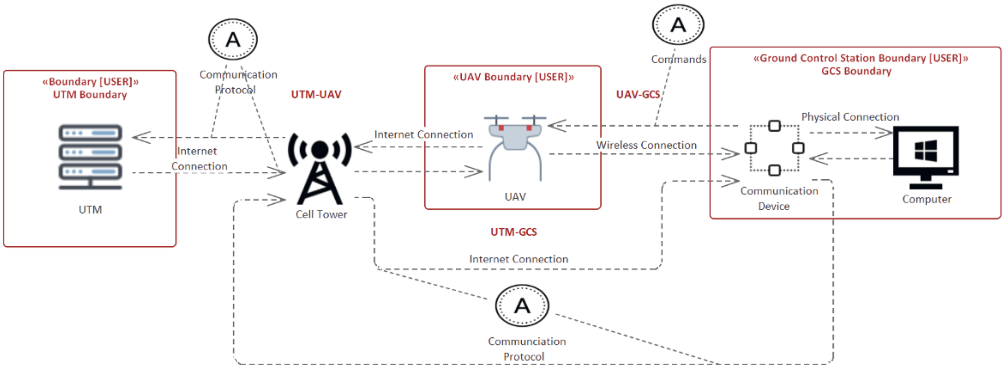


Fig. 2: A diagrammatic representation of components and communications in a U-space framework.

single value expressed by the equal sign (”=”) or a range of many values represented by IN [”value1”, ”value2”, ”etc.”). The grammar can also describe connectors that cross boundaries in order to investigate the flow between borders. The ”crosses filter” checks whether connectors cross an element or a boundary. Additionally, an ”asset filter” is defined in the grammar to describe that the connection has an asset, as shown in figure 2 on the flow from GCS to the UAV.

Listing 1: Part of the ThreatGet formal grammar, the full-description of the grammar is discussed in [26]

```
connector_pattern -> CONNECTOR (type_filter)?
    {source_filter & target_filter (&
    connector_filters)}
source_filter -> SOURCE element_pattern
target_filter -> TARGET element_pattern
element_pattern -> (element_pattern (|
    element_pattern)+) ELEMENT (type_filter)?
    ({element_filters})?
element_filters -> element_filters (&
    element_filters)+ ( element_filters (|
    element_filters)+ ) tagged_value_filter
tagged_value_filter -> "key" = "value" "key"
    != "value" "key" IN ["value" (,
    "value")*] "key" NOT IN ["value" (,
    "value")*]
connector_filters -> connector_filters (&
    connector_filters)+ ( connector_filters
    (| connector_filters)+ )
    tagged_value_filter crosses_filter
    asset_filter
crosses_filter -> CROSSES element_pattern
    CROSSES boundary_pattern
asset_filter -> HOLDS asset_pattern HOLDS NO
    asset_pattern
```

ThreatGet classifies threats according to the STRIDE model. STRIDE is the abbreviation of the **S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, and **E**levation of Privilege [27]. It was invented in 1999 and adopted by Microsoft [4] in 2002 [28]. Each classified category of threats violates a particular security property. The threat categories are discussed in [29] as follows:

- **Spoofing**: Get unauthorized access by violating **authentication**.
- **Tampering**: Modify or damaging data in an unauthorized way by violating **integrity**.
- **Repudiation**: Denying an activity that a legal/illegal user by violating **non-repudiation**.
- **Information Disclosure**: An undesirable manner could reveal data by violating **confidentiality**.
- **Denial of Service**: An unauthorized action leading to the unavailability of a specific service, system, or application by violating **availability**.
- **Elevation of Privilege**: A restricted authorized user could claim a higher privilege than they hold by violating **authorization**.

Afterwards, we perform the threat analysis using ThreatGet

[4] www.microsoft.com

to identify potential threats in the UAV scenario as described in Figure 2, initially assuming that the assets do not hold any security properties, i.e. are not protected by any security mechanism.

The tool detects 35 threats that have negative consequences against multiple units in our UAV example. These threats are classified according to the STRIDE model, as discussed previously. There are six threats classified as Spoofing attacks, where 16 threats violate the integrity of data. Hence, these threats are classified as Tampering. Four other threats are categorized as Information Disclosure. Besides, only one threat is targeting Repudiation, and another one Elevation of Privilege; seven threats are classified as denial of service attacks. ThreatGet also automatically performs risk evaluation to estimate the severity level for each identified threat, as discussed in Section II-D.

D. Risk Evaluation and Security Target Estimation

It is essential to estimate the ST for each zone and conduit in order to be able to select the most applicable security requirements that address existing security issues. Therefore, we estimate the security target for each zone and conduit according to the risk severity of threats that ThreatGet identifies. ThreatGet calculates the overall risk of the whole UAS model by estimating the risk severity of each identified threat. This calculation is based on parameter values impact and likelihood.

- **Impact**: Estimates the harm that could be caused by a particular threat. Furthermore, in this work, we propose five scenarios that could be affected by cyberattacks.
  - **UAV Safe Operation**: The degree of harm caused by a threat against UAV's safe operation.
  - **Operation Stop Working**: The impact degree of a threat can affect the operation of a UAV.
  - **Financial Impact**: This scenario reflects the impact degree against the financial assets.
  - **Breach Data Integrity**: This scenario expresses the degree of data integrity violation by attackers.
  - **Breach Data Confidentiality**: This scenario represents the degree of a data confidentiality breach.
- **Likelihood**: Estimate the probability of a threat that could occur.

ThreatGet's impact values are determined based on four degrees that reflect the severity of the threat's impact. The lowest degree of impact is **Negligible** (i.e., 1), while the maximum degree of impact is **Severe** (i.e., 4). We then estimate the Mean and determine the impact degree for each threat based on the five scenarios as previously discussed. ThreatGet also uses four degrees to express the likelihood values. The lowest degree is ”Very Low” (i.e., 1), where the maximum degree is ”High” (i.e., 4). According to this estimation, ThreatGet estimates the risk severity for each identified threat based on the following formula:

$$Risk\ Severity = Impact * Likelihood$$

TABLE I: Security target analysis of GCS security zone and UTM-GCS conduit based on ThreatGet's findings

Threats	GCS	UTM-GCS	Risk Severity	STRIDE	Violation
T1	X		1	I	Confidentiality
T4	X		1	I	Confidentiality
T8	X		4	T	Integrity
T9	X		3	D	Availability
T11	X		3	D	Availability
T12	X		2	R	non repudiation
T13	X	X	3	D	non repudiation
T14	X		2	T	Integrity
T19	X		2	T	Integrity
T20	X	X	2	T	Integrity
T21	X		2	T	Integrity
T22	X		2	S	Authentication
T23	X		2	S	Authentication
T24	X	X	2	T	Integrity
T25	X		4	E	Authorization
T26	X		3	T	Integrity
T27	X		2	T	Integrity
T28	X		3	T	Integrity
T29	X		1	S	Authentication
T30	X		1	D	Availability
T32	X		1	S	Authentication
T34	X	X	1	D	Availability
T35	X		2	S	Authentication
ST of GCS	Level 4				
ST of UTM-GCs	Level 3				

Estimating the risk severity for each threat helps to determine the security target that needs to be achieved for each security zone and conduit. Therefore, we analyze the risk estimation process results to define the risk severity associated with each security zone (i.e., UAV, UTM, and GCS) and conduit (i.e., UTM-UAV, UTM-GCs, and UAV-GCS). Table I illustrates all threats that are identified by ThreatGet, within the GCS zone and UTM-GCS conduit.

The table contains 23 threats that affect the GCS security zone; each threat is classified according to the STRIDE category, as discussed in Section II-C. Similarly, at the UTM-GCS conduit, this conduit is affected by four threats. We estimate the ST for each zone and conduit based on the threat's highest risk severity. The highest risk severity for the GCS is 4, while the highest severity for UTM-GCs is 3. Therefore, the GCS's ST is assumed to be level 4, whereas the UTM-GCS's ST is level 3. The table also describes threats according to the STRIDE categories and its violation based on these categories. For example, **T8** is classified as Tampering category, which violates the integrity of data in unauthorized behaviour.

Section II-E, discusses how the finding of ThreatGet could assist in applying IEC 62443 based on the ST and violation of security properties.

E. Apply Security Requirements and Map FRs with STRIDE

In this work, we propose using the IEC 62443 [7] to provide a complete cybersecurity framework for addressing existing cybersecurity issues. Security requirements are classified according to their level of capability, referred to as Security-Level Capability (SL-C). This level describes the security level that system units should fulfill without additional measures [30]. Each security requirement is defined in the IEC 62443 with a range of capability levels varying from 1 (i.e., casual exposure) to 4 (i.e., sophisticated means). The standard describes security requirements into FRs. These requirements are discussed in [30], as follows:

- **FR1 - Identification and Authentication Control (IAC)** Supports authentication and manages cybersecurity issues relevant to spoofing activities.
- **FR2 - Use Control (UC)** Provides authorization and handles cybersecurity issues related to violation of system/software privigle.
- **FR3 - System Integrity (SI)** Supports data integrity, and handle issues related to tampering activities.
- **FR4 - Data Confidentiality (DC)** It intends to prevent unauthorized data access either on communication channels or stored.
- **FR5 - Restricted Data Flow (RDF)** It restricts unnecessary data flows by building zones and conduits, which helps in limiting the propagation of cyberattacks.
- **FR6 - Timely Response to Events (TRE)** Supports handling security concerns associated with multiple forms of repudiation attacks.
- **FR7 - Resource Availability (RA)** Supports handling multiple forms of resource availability attacks.

The security requirements shall be selected to protect system assets against any form of cyberattack and address existing security issues. According to the outcomes that are presented in Table I, we introduce a mapping procedure that enables a direct mapping between security requirements (defined in terms of FRs) and threats (defined in terms of STRIDE), as illustrated in Figure 3. This procedure clarifies how to apply IEC62443-4-2 security requirements in the UAS domain to develop secure civil applications.

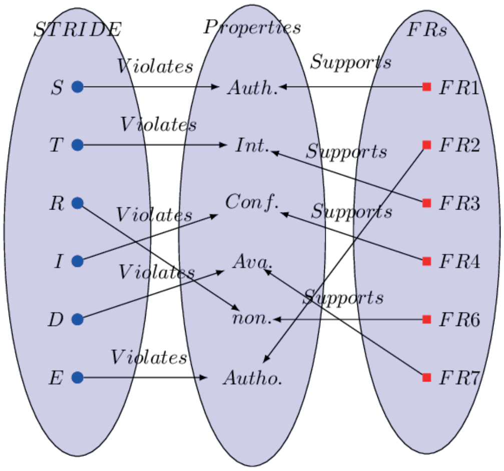


Fig. 3: The proposed mapping strategy between STRIDE and FRS according to the common security properties
Abbreviation: Auth. Authentication, Int. Integrity, Conf. Confidentiality, Ava. Availability, non. non-repudiation, and Autho. Authorization

The STRIDE categories are specified as the primary categories of identified threats on the left-hand side. On the right-hand side is the list of all FRs that classifies the security requirements of the IEC 62443 security standard. All security properties that are violated by threats are defined in the middle list. Also, these properties are defined in security requirements in terms of FRs. Furthermore, due to threat's violating security properties, relevant security requirements shall be selected to address existing security issues. In addition, based on ThreatGet findings, we will be able to select a set of security requirements depending on their security capabilities (i.e., SL-C). These capabilities should equal each threat's risk severity to achieve the main ST for each security zone and conduit.

Therefore, this mapping technique is considered a one-to-one approach to provide a precise and direct linkage from security issues (i.e., threats) to security solutions (i.e., security requirements).

III. Conclusion and Future Work

We proposed a standard-based procedure based on IEC 62443 to be integrated in the UAS-domain for addressing potential threats. We employ ThreatGet as a threat modelling tool to assist in this process. We define security zones/conduits and define the main system's assets. Then we perform the risk analysis using ThreatGet for analyzing, detecting, and prioritizing security issues of a system design. The tool defines a set of threats and classifies them using the STRIDE model. Then the tool estimates the severity level for each threat based on the values of impacts and likelihoods. We then utilize these findings to describe a mapping strategy to select a proper set of security requirements for addressing existing threats. The proposed mapping strategy is based on selecting a set of security requirements according to their capabilities to fulfil the main security goal. In addition, describe a clear understanding between FRs, that support security properties and the STRIDE model according to the violation of common properties.

Our future work will include developing a mathematical model that estimates the security achieved after applying security requirements. That helps to guarantee the achieved level is equal to the security target level and ensure the correctness of the applied security requirements.

IV. Acknowledgement

This work is done in the LABYRINTH project, which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 861696.

References

[1] S. J. Undertaking, "European drones outlook study. unlocking the value for europe," *SESAR, Brussels*, 2016.
[2] CORUS, "U-space concept of operations," SESAR, Tech. Rep., 2019. [Online]. Available: https://www.sesarju.eu/U-space
[3] L. Project, "Ensuring drone traffic control and safety," http://labyrinth2020.eu/the-project/, 2017, (Accessed on: September 18, 2021).
[4] "ISO/IEC 27000 – key international standard for information security revised," https://www.iso.org/cms/render/live/en/sites/isoorg/contents/news/2018/03/Ref2266.html, (Accessed on: September 10, 2021).
[5] "ISO 15408, information technology - security techniques - evaluation criteria for IT security (Common Criteria)," 2009.
[6] ISA, "The 62443 series of standards: Industrial automation and control systems security," no. 1-4, 2018.

[7] IEC, “Security for industrial automation and control systems - part 4-2: Technical security requirements for IACS components,” International Standard, Tech. Rep., Feb. 2019.

[8] AIT, “Threatget - threat analysis and risk management,” https://www.threatget.com, 2019, acessed: 2021-07-17.

[9] SECOPS, “Security concept for drone operations (SECOPS),” SESAR, Tech. Rep., 2019. [Online]. Available: https://www.sesarju.eu/projects/secops

[10] SJU, “SecRAM 2.0 – security risk assessment methodology for SESAR 2020,” SESAR, Tech. Rep., 2017. [Online]. Available: https://www.sesarju.eu/projects/secops

[11] JARUS, “Jarus guidelines on specific operations risk assessment (sora),” JARUS, Tech. Rep., 2019. [Online]. Available: http://jarus-rpas.org/content/jar-doc-06-sora-package

[12] SESAR, “Safety reference material, edition 4.0,” SESAR, Tech. Rep., 2016. [Online]. Available: https://www.sesarju.eu

[13] FAA, “Security considerations for operationalization of utm architecture,” Tech. Rep., 2021. [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/20210112_-_final_upp2_security_analysis_0.pdf

[14] “ISO/IEC 27005: Information technology — security techniques — information security risk management – second edition,” 2011.

[15] CSRC, “NIST Special Publication (SP) 800 Series,” Tech. Rep., 1990. [Online]. Available: https://csrc.nist.gov/publications/sp800

[16] J. Valente and A. A. Cardenas, “Understanding security threats in consumer drones through the lens of the discovery quadcopter family,” in *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, 2017, pp. 31–36.

[17] B. Rao, A. G. Gopi, and R. Maione, “The societal impact of commercial drones,” *Technology in Society*, vol. 45, pp. 83–90, 2016.

[18] A. Y. Javaid, W. Sun, V. K. Devabhaktuni, and M. Alam, “Cyber security threat analysis and modeling of an unmanned aerial vehicle system,” in *2012 IEEE Conference on Technologies for Homeland Security (HST)*. IEEE, 2012, pp. 585–590.

[19] G. L. Lattimore, “Unmanned aerial system cybersecurity risk management decision matrix for tactical operators,” NAVAL POSTGRADUATE SCHOOL MONTEREY CA MONTEREY United States, Tech. Rep., 2019.

[20] M. R. Manesh and N. Kaabouch, “Cyber-attacks on unmanned aerial system networks: Detection, countermeasure, and future research directions,” *Computers & Security*, vol. 85, pp. 386–401, 2019.

[21] E. K. et al., “D2.3 Architecture Requirements and Definition (v2),” afarcloud deliverable, Tech. Rep., February 2020. [Online]. Available: http://www.afarcloud.eu/wp-content/uploads/2020/04/D2.3-Architecture-Requirements-and-Definition-2.0_VFINAL.pdf

[22] Sander Walters, “How to set up a drone vulnerability testing lab,” https://medium.com/@swalters/how-to-set-up-a-drone-vulnerability-testing-lab-db8f7c762663, 2016, (Accessed on: September 12, 2021).

[23] T. Macaulay, “The 7 deadly threats to 4g: 4g lte security roadmap and reference design,” *Accessed on: September 16, 2021*, vol. 25, p. 2017, 2013.

[24] UNECE, United Nations Economic Commission for Europe, “CSOTA ad hoc ”threats 2”,” https://wiki.unece.org/download/attachments/45383725/TFCS-ahT2-06%20%28Chair%29%20Table%20on%20CS%20threats%20-%20changes%20agreed%20by%20ahT2%20-%20non-cleaned%20up.xlsx?api=v2, 2017, (Accessed on: September 18, 2021).

[25] K. Kotapati, P. Liu, Y. Sun, and T. F. LaPorta, “A taxonomy of cyber attacks on 3g networks,” in *International Conference on Intelligence and Security Informatics*. Springer, 2005, pp. 631–633.

[26] AIT, “Welcome to the threatget documentation,” https://documentation.threatget.com/21.06/, 2019, acessed: 2021-07-17.

[27] A. Shostack, *Threat modeling: designing for security*. Wiley, 2014, OCLC: ocn855043351.

[28] N. Shevchenko, “Threat modeling: 12 available methods,” https://insights.sei.cmu.edu/sei_blog/2018/12/threat-modeling-12-available-methods.html, 2018, accessed on: September 20, 2021.

[29] M. Abomhara, M. Gerdes, and G. M. Køien, “A STRIDE-based threat model for telehealth systems,” *NISK Journal*, pp. 82–96, 2015.

[30] International Electrotechnical Commission, “IEC 62443-3-3: Industrial communication networks – network and system security – part 3-3: System security requirements and security levels,” 2013.

IAIDO: A Framework for Implementing Integrity-Aware Intelligent Data Objects

Eric Davis
Galois, 22203 Virginia, United States
eric.davis@galois.com

***Abstract*—Growing reliance on automated reasoning, machine learning, and machine-aided decision making has lead to serious vulnerabilities in the area of data-integrity. The trustworthy and reliable operation of next-generation data-driven systems and the infrastructure which manages this data will require effective and scalable solutions to the growing threat of faults due to data-integrity. In this paper we discuss the concept of data-integrity, outline threats to data-integrity, and introduce the notion of Integrity-Aware Data Objects which utilize the concepts of polymorphism, subsumption, composition, association, and aggregation to build a system of inheritance to improve data-integrity for large-scale data sets with shared provenance, representations, and types. We further extend the notion of these data objects by adding intelligence in the form of learned constraints, rules, and classifiers which are inherited by data-objects to improve tolerance of data-integrity errors. We implement these Integrity-Aware Intelligent Data Objects as the IAIDO framework, and demonstrate this novel approach using real data on nutrition information, providing examples of real data-integrity faults in the USDA's National Nutrient Data Base for Standard Reference Release 28, and in crowd sourced data. We demonstrate high rates of data-integrity faults in crowd sourced data, with nearly 27% of our data failing one or more SMT-based constraints. Similarly, in federally published data we find nearly 10% of data published by the USDA is non-compliant, and features data-integrity faults.**

***Index Terms*—component, formatting, style, styling, insert**

I. Introduction

Data-driven applications, science, and processes rely on the collection of large volumes of data at high velocities, with additional challenges presented due to the variety of data collected, and the strong requirements for veracity, the so-called 4 V's of big data [1]. As organizations and individuals become more reliant on the trustworthy outcomes of data-driven processes, they also increase their risk and exposure to faults, errors, and failures due to low-integrity data [2]. Crowd-sourced data holds the promise of rapid data-acquisition, ease of use, and low-cost for deployment and implementation and has seen growth in areas such as disaster relief [3], emergency response [4], and even healthcare [5] and clinical trials [6]. The use of crowd-sourced data becomes a liability, however when it comes to data veracity [7] as it reduces accountability [8], [9], and the ability to build trust due to the provenance of the data [10]. While crowd-sourced data is particularly vulnerable to problems with veracity, data cleaning as a whole has long been a core unsolved problem with data warehousing [11], [12]. When data is unreliable, or worse still, maliciously injected to drive machine-based reasoning towards incorrect outcomes, the integrity of the entire system is called into question.

In order to support reliable and trustworthy next-generation data applications, processes, and scientific techniques must be developed to account for integrity as a core system-design principle, not as an after-thought. Next-generation systems will have to fight a constant battle against low-integrity data seeking to pollute their archives, and we must move from a more reactive, bespoke approach to core frameworks which support reasoning about data-integrity in a formal manner.

We advance the hypothesis that data-integrity can be improved by encapsulating all data ingested into a data-warehouse, application, or experiment in an object-framework that allows the assignment of formal, context-sensitive, *types* and the inclusion of both data, in the form of *object fields*, and procedures which can act upon that data, known as *object methods*, to associate data with algorithms which reason about the integrity of the data held within the object. This paradigm is based on the following assumptions: (1) That data in most large-scale systems belongs to a, often incomplete, ontology or taxonomy of related data in which similarly *typed* data objects exist within the system. (2) That knowledge about the integrity of some data objects can be leveraged to improve knowledge about the integrity of other data objects that are ontologically related through subsumption (`is-a`, or subtype relationships) and composition (`has-a` relationships). We do not argue that these assumptions hold universally, merely that they hold often enough to be of value when reasoning about data-integrity and improve our ability to identify and remove low-integrity data, and maliciously injected data from our system.

Our contributions are summarized as follows:

- We introduce a formal semantics of object inheritance and provide formal definitions within a type environment for what it means for a data-object to subsume another, or be composed with another.
- We use our formal semantics to implement the IAIDO Framework and Quarantine prototype using a novel extension of the Javascript Object Notation and python implemented object methods that are assembled by our IAIDO Framework parser.
- We apply our IAIDO Framework to real data from a case study investigating the relationship of food nutritional data with blood glucose levels. This data was found to contain many low-integrity points, and furthermore

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_8

the low-integrity of this data impacts machine-learning applied to the blood glucose model.

- We demonstrate the efficacy of our IAIDO framework and show the results, experimenting on a collection of 610 partially labeled data.
- **We identify a case of a major food manufacturer mislabeling product data that was previously unknown.** We do so through the implementation of FDA constraints on food labeling as a user-defined integrity constraint.

II. Related Work

Formally the issue of trustworthiness of data within a system hinges on the dependence on that data to inform normal service. We require **trustworthy** or **high integrity** data when the dependability of our system relies on the dependability of the data in order to deliver correct service. Thus trustworthy data is data on which correctness is required, and accepted by the system for correct service. [13], [14]

Fundamentally the problem of addressing trustworthy data-systems through data integrity evaluation shares more with notions of trustworthiness from the security community. The assurance that a system performs as expected even in the presence of hostile attacks from outsiders, or insiders, environmental disruptions, and human/operator error. [2], [8], [9], [15] It also has elements of data veracity, a term introduced when talking about the four V's of big data. In this sense data with high integrity should exhibit consistency, or statistical reliability. In cases where trustworthiness on the basis of data origin and collection cannot be established, we need to instead develop formal ontologies of data relationships where we can repudiate low integrity data on the basis of inconsistency. [7]

The related work focuses on solving this problem using multi-tenant environments with cooperation for security [16], [17] in which central authorities serve as gate keepers for data, primarily looking at data-integrity as a problem of access control. In the literature data provenance is used to help with policy binding for access. These sorts of solutions, however do not help in situations of partially or wholly crowd sourced data where provenance and trust can't be assured [18], [19]

III. Preliminaries

In this section we provide a formal definition for data-objects, and explain the semantics of inheritance using subsumption and composition. Our formal model of objects is based on an extended Kripke structure [20] and provides two main features *encapsulation* and *inheritance*. The members of an object are not unrelated, but instead collectively share responsibility for representing the state of an object (object fields), or changing the state of an object (object methods).

Informally, an object is a collection of components which we distinguish as *fields* and *methods*. For our purposes we consider each field to be an object itself, and associate the set of fields which values drawn from some set. We characterize the set of all of an object's fields and their current values, with the state of the object at some moment in time. Methods can be considered as functions or procedures which make use of an object and its fields to yield a new state for the object after they are invoked. Formally we define an object as an extended Kripke structure [20] represented as a 5-tuple: $o = (F, f_0, Q, M, \delta)$, consisting of

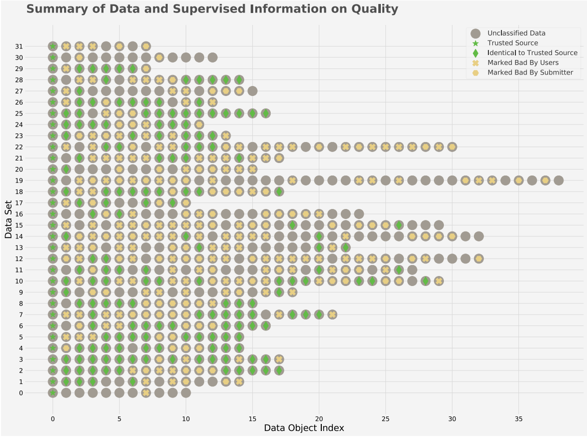


Fig. 1: Summary of the 610 data points in 32 data sets used for our experiments. Data in column 0, marked with a green star, represent manufacturer data. Data marked with a green diamond are identical to manufacturer data. Data marked with a yellow "X" were marked as suspected low-integrity by our subjects. Data marked with a yellow hexagon were identified as low-integrity by whomever entered the data.

- a finite set of n fields, $F = \{f_0, f_1, \ldots, f_{n-1}\}$ each of which is an object itself.
- The special field f_0 which represents *self*, and corresponds to the object itself.
- the state of the object, $Q \to \mathbb{N}$ which represents the current state of all fields of the object.
- a finite set of p method labels $M = \{m_0, m_1, \ldots, m_{p-1}\}$.
- a transition function $\delta : Q \times M \to Q$ which represents the results of method invocation of a given method $m_i \in M$ when the object is in a state $q \in Q$ representing some set of values assigned to the fields of the object and transition to a new state $q' \in Q$ as $\delta(q, m_i) \to q'$

We use the shorthand $o.X$ to represent some set from (F, Q, M, δ) in an object o, and $o.x_i$ to represent an element of one of the sets (F, Q, M, δ) in an object o providing for method invocation, or field selection using a simpler syntax. We allow for method override (or update) using the syntax $o.m_i \Leftarrow (m'_i, \delta'|_{m'_i})$ with

$$o.m_i \Leftarrow (m'_i, \delta'|_{m'_i}) \quad \triangleq \quad o.\delta = (o.\delta|_{M \smallsetminus m_i} \oplus \delta'|_{m'_i}), \tag{1}$$

$$o.M = (o.M \smallsetminus o.m_i) \cup m'_i \tag{2}$$

Intuitively this represents overiding the method $o.m_i$ with a new method m'_i by replacing it in $o.M$ and updating the transition function $o.\delta$ with the new transition for $o.m'_i$ using function restriction $(|)$ and extension via overriding union $(\oplus)$ [21], [22]. Method updates are used to replace or override existing methods, which may have been inherited from another type which our type subsumes.

In addition to overriding and updating methods, we can also remove or add methods. We use the notation $o.\epsilon$ to represent the null method, allowing the addition of new methods with:

$$o.\epsilon \Leftarrow (m_i', \delta'\big|_{m_i'}) \quad \triangleq \quad o.\delta = (o.\delta \oplus \delta'\big|_{m_i'}), \tag{3}$$

$$o.M = o.M \cup m_i' \tag{4}$$

Or method removal from an object using:

$$o.m_i \Leftarrow (\epsilon, \epsilon) \quad \triangleq \quad o.\delta = o.\delta\big|_{M \setminus m_i}, \tag{5}$$

$$o.M = o.M \setminus o.m_i \tag{6}$$

Field replacement, addition, and removal semantics follow similarly,

$$o.f_i \Leftarrow (f_i') \quad \triangleq \quad o.F = (o.F \setminus o.f_i) \cup f_i' \tag{7}$$

$$o.\epsilon \Leftarrow (f_i') \quad \triangleq \quad o.F = o.F \cup f_i' \tag{8}$$

$$o.f_i \Leftarrow (\epsilon) \quad \triangleq \quad o.F = o.F \setminus o.f_i \tag{9}$$

$$\tag{10}$$

with the restriction that the self field, f_0, cannot be removed or replaced. The self field, f_0 is special, and its value is always given by definition as:

$$o.f_0 \triangleq o$$

Our objects exist with a typing environment E [23] where each object o has a type τ written as $o : \tau$. The notation for inference involves sequents, or inference rules of the general form

$$\frac{E \vdash o_0 : \tau_0 \qquad \cdots \qquad E \vdash o_{k-1} : \tau_{k-1}}{E \vdash o : \tau}$$

where the rule, or sequent $E \vdash o : \tau$ indicates object o has type τ in the environment E [24]. This allows us to indicate subclass relationships using the subsumption relation <:, such as $A <: B$ which indicates type A subsumes, or is a subclass type B (intuitively A `is-a` B) [25], [26],

$$\frac{E \vdash a : A \qquad E \vdash b : B \qquad E \vdash A <: B}{E \vdash a : B} \tag{11}$$

and to indicate composition using $A \relbar\!\!\bullet B$ to indicate A is composed with B, (intuitively B `has-a` A),

$$\frac{E \vdash a : A \qquad E \vdash b : B \qquad E \vdash A \relbar\!\!\bullet B}{a \in b.F} \tag{12}$$

Subsumption allows us to define new types either partially, or wholly through implicit means. Given some type $E \vdash o_i : \tau_i$ and $E \vdash o_j : \tau_j$ where $E \vdash \tau_i <: \tau_j$, if no override $o_j.m_k \Leftarrow (m_k', \delta'\big|_{m_k'})$ has been defined then $o_j.m_k \triangleq o_i.m_k$, with similar rules for fields. By the same token if $E \vdash o_i : \tau_i$ and $E \vdash o_j : \tau_j$ where $E \vdash \tau_i \relbar\!\!\bullet \tau_j$, there exists some field $f_l \in o_j.F | E \vdash f_l : \tau_i$. Intuitively if one object's type subsumes another, that type inherits all the same methods and fields implicitly unless replaced or removed, and if one object is composed with another, the second object contains the first object as a member field.

As part of our IAIDO framework prototype, we implemented a basic quarantine system that operates on a rules basis on objects in our system. We established our quarantine protocols to be risk-averse, as in our case study we found no cases in which incorrect identification of manufacturer data by machine-learning methods occurred for the same point for any two intelligent integrity checks, leading to a voting strategy for intelligent checks. Conversely we use the user-defined constraints from the FDA as a hard reject, because if a product is labeled improperly, these checks will detect the inconsistency which means the data supplied cannot possibly comply.

IV. Experimental Evaluation

We conducted our experiments on nutritional data submitted to MyFitnessPal, and recorded in the diaries of subjects being studied in order to model the impact of meals on blood glucose over a period of one year. We selected a subset of 32 different food items, unified by UPC (universal product code) data, and MyFitnessPal database resource number. For each separate database entry we had our subjects mark the entry if they suspected the entry to be of low-integrity. We additionally added manufacturer provided data, and marked those entries which were identical to the manufacturer's provided data. Further more some entries were marked as having low-integrity by whomever entered the data, with key phrases such as "(Net carbs)" indicating that the provided data had been modified on the basis of some dieting trick, or "(Carbs removed)", and similar labels.

A. User-Defined Experimental Constraints

We applied two user-defined data integrity constraints, one at the level of `carbohydrate_info` based on FDA guidelines which state that the total grams of carbohydrates must be greater than or equal to the sum of the component carbohydrates (fiber and sugar in our data). This was an important data integrity constraint for our motivating case study as the total carbohydrate content, both fiber and non-fiber sources, was needed for our blood glucose estimation, and provides healthcare providers with important information on the glycemic impact of foods consumed.

Our second user-defined data integrity constraint was derived from the FDA rules on nutrition labeling and rounding guidelines [27] which provides a formula for labeling the kilocaloric content (colloquially referred to simply as calories) of food on the basis of macro nutrient composition. The FDA defines its nutrient rules using a complex set of rounding guidelines which provides that if the total kilocalories are < 5, or the total grams of a macro nutrient are $< 0.5g$ they should be expressed as zero. For products with caloric content $5 < x \leq 50$ kilocalories, the kilocalories should be rounded to the nearest 5 kilocalorie increment. For caloric content $x > 50$ kilocalories, they should be rounded to the nearest 10 kilocalorie increment. Macronutrients should likewise be rounded to $0.5g$ increments when total content is $0.5g < x < 5g$, and to the nearest one gram increment when total macronutrient content is $x \geq 5$ grams. These were implemented as constraints as a Python 3 module imported as a method for our data objects as shown in Figure 2.

B. `food_info` Intelligent Constraints

We applied two machine-learned constraints that were inherited by call subsumptions of the `food_info` class (and thus encompassed all classes). We trained both a linear, and non-linear regression of kilocalories on all other fields in the `food_info` class on when unmarked by our users to attempt and derive a non-user supplied constraint similar to the FDA supplied constraint, but to also look for other possible violations beyond macro-level kilo-caloric content. A random subset of 30% of our unlabeled data was submitted as training data, with k-fold cross validation techniques employed [28], [29].

C. `food_info` Sub-Class Intelligent Constraints

Our last intelligent constraint consisted of a random forest [30] over all features for a two-class classifier to attempt to mimic the classification procedure conducted by our users when marking data as "suspected low-integrity". A random subset of 30% of our labeled data was submitted as training data, with k-fold cross validation techniques employed [28].

D. Experimental Results

When we applied our user-defined constraints, we recovered a number of violations immediately, some of which were labeled by our users, or those entering the data, some of which were not, Figure 3 summarizes these results. A total of 93 data entries violated the FDA carbohydrate constraint in our data set. A total of 143 data entries violated the FDA kilo-caloric constraint in our data set. Most interestingly, **data provided by the manufacturer for item number 3 was found to violate the FDA's constraints on labeling**. These results were hand checked and confirmed after appearing in quarantine.

Applying intelligent constraints resulted in highly accurate results, which in general improved as more data sets were added (each contributing 30% new training points so the classifier improved over time). Once all 32 data sets were included, the classifier mis-predicted only 8 points while correctly identifying 138 points as low-integrity. Of the 8 points that were misclassified, they were misclassified by only one of the two regression methods in all but one case, that of data set zero. Both methods agreed on the 138 other points correctly identified as low-integrity. The one outlier experiment, data set zero, was a singular case in our case study.

The random forest classifier proved exceptionally good at identifying why users tended to mark data as "suspected low-integrity". Results for the `questbar` class are shown above in isolation to help better visualize and understand these results. This classifier was not shared by all `food_info` classes, but only the `questbar` <: `food_info` type. Each type had its own random forest classifier with similar results. For the `questbar` type, 109 suspected low integrity data points were identified, including 42 not identified by other intelligent, or user-defined constraints. Similar results were achieved with the other classes, which are omitted for space reasons. There were no cases of the random forest classifier mis-predicting data which matched the manufacturer's supplied data.

E. Reasoning with Quarantine

Figure 4 shows a summary of the data our quarantine prototype used for reasoning. In general the darker the square, and higher the indicated number, the more sure the quarantine system is that the data lacks integrity. The one exception is row 3, wherein the single violations account for violations of the user supplied kilocaloric constraints from the FDA, and are removed due to violation of federal standard.

Subclass indicators, such as the random forest, showed great promise. Across all of our subclass classifiers, 0.79 of user marked data was recalled using only 30% of the marking as training data, 0.00 of data which matched the manufacturer was marked as "suspected low-integrity", and 0.45 of the unlabeled data which did not match the manufacturer was also identified as more closely matching the "suspected low-integrity" data than manufacturer data.

V. Conclusions

We have presented a new semantics for data objects which provides rules for formal inheritance in data objects of both fields, and integrity-aware methods under the rules of subsumption and composition to support the creation of Integrity-Aware Intelligent Data Objects as part of our IAIDO framework. We have implemented a prototype of our framework using an extension of the Javascript Object Notation, and Python, and included as part of our framework a prototype quarantine system based on empirical results with real data from a case study in personalized health data which suffers from data-integrity problems. Our method has proven effective in the identification and elimination of large portions of low-integrity data from our data set, ensuring data is more representative and accurate for machine-driven reasoning and data scientific applications.

In our case study we additionally uncovered a case of a manufacturer violating federal labeling requirements for quite some time that has gone unnoticed by federal regulatory bodies. This provides increased motivation for the further development and deployment of data-integrity aware systems to help enforce important standards for honest and safe labeling of products. While it is unclear how long this manufacturer has been labeling their products in violation of FDA regulations, we found product labels from over a decade ago with these incorrect labels. It is impossible for us to tell, from the data we have access to, whether the mislabeling is

$$kcal + \epsilon_{kcal} = (f + \epsilon_f) * 9 + (p + \epsilon_p) * 4 + ((c + epsilon_c) - (df + \epsilon_{df})) * 4 \tag{13}$$

$$\big((kcal = 0) \rightarrow (0 \leq \epsilon_{kcal} \leq 5)\big) \wedge \big((0 < kcal \leq 50) \rightarrow (-5 \leq \epsilon_{kcal} \leq 5)\big) \wedge \big((50 \leq kcal) \rightarrow (-10 \leq \epsilon_{kcal} \leq 10)\big) \tag{14}$$

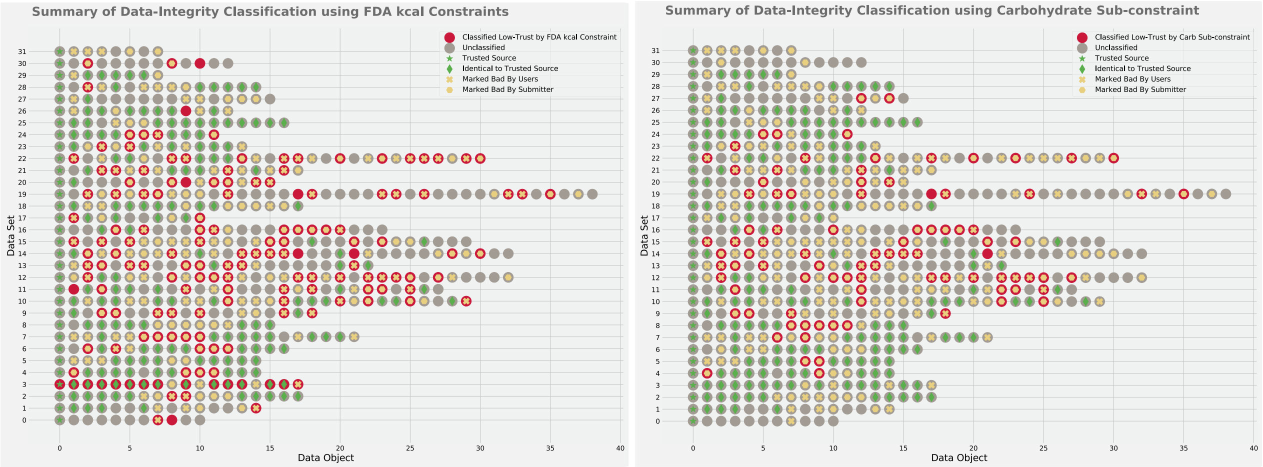
$$\big((f \leq 0.5) \rightarrow (-f \leq \epsilon_f \leq f)\big) \wedge \big((0.5 < f < 5) \rightarrow (-0.5 < \epsilon_f < 0.5)\big) \wedge \big((5 \leq f) \rightarrow (f - \lfloor f \rfloor \leq \epsilon_f \leq \lceil f \rceil - f)\big) \tag{15}$$

$$\big((p \leq 0.5) \rightarrow (-p \leq \epsilon_p \leq p)\big) \wedge \big((0.5 < p < 1) \rightarrow (-1 < \epsilon_p < 1)\big) \wedge \big((1 \leq p) \rightarrow (p - \lfloor p \rfloor \leq \epsilon_p \leq \lceil p \rceil - p)\big) \tag{16}$$

$$\big((c \leq 0.5) \rightarrow (-c \leq \epsilon_c \leq c)\big) \wedge \big((0.5 < c < 1) \rightarrow (-1 < \epsilon_c < 1)\big) \wedge \big((1 \leq c) \rightarrow (c - \lfloor c \rfloor \leq \epsilon_c \leq \lceil c \rceil - c)\big) \tag{17}$$

$$\big((df \leq 0.5) \rightarrow (-df \leq \epsilon_{df} \leq df)\big) \wedge \big((0.5 < df < 1) \rightarrow (-1 < \epsilon_{df} < 1)\big) \wedge \big((1 \leq df) \rightarrow (df - \lfloor df \rfloor \leq \epsilon_{df} \leq \lceil df \rceil - df)\big) \tag{18}$$

Fig. 2: The background theories derived from FDA guidlines provided in “Labeling & Nutrition Guidance Documents & Regulatory Information” used to model the kilocaloric content of food on the basis of macronutrient composition.



(a) Summary of violations of the FDA kilocaloric constraints. Data points with a red background were found in violation of the contraint.

(b) Summary of violations of the FDA carbohydrate constraints. Data points with a red background were found in violation of the contraint.

Fig. 3: Violations found of the FDA kilocaloric and carbohydrate user-defined constraints.

due to misrepresentation of caloric content, or macro-nutrient content, but it is clear that the label is inconsistent with FDA regulations and the error has gone uncaught.

There was a single instance of potentially good data being discarded from our set, from data set zero. In data set zero none of the data matched the manufacturers label. This was a singular incident for our test data, and we do not treat its quarantine as necessarily a bad thing. While manufacturer data for this product does match FDA guidelines, for some reason all crowd-sourced provided failed to match the manufacturer data. The item in question was rather generic (a slice of bread from a name brand manufacturer) and so may often be input improperly.

VI. Future Work

In order to support time-series data integrity checks, which include constraints on rates of reasonable change, and relation between points both learned from the data, and supplied by the user, additional semantics must be added to IAIDO for aggregation relationships between data objects. Aggregation extends the current `has-a` relationship modeled by composition to include when the a set of related data types are stored in an aggregate composed relationship in which position in the aggregate may be contextually important. We are currently exploring both the semantics, implementation, and experimental considerations of aggregate relationships using time-series data recorded for heart-rate, position, and other health-indicators for our study groups.

Currently all objects are defined by the user, and when new data is entered into the system, it must be typed. We are investigating means of conducting type-inference on data drawn from a finite set of known types to further automate the process, based both on user-defined constraints, and learned type rules similar to those employed for data-integrity.

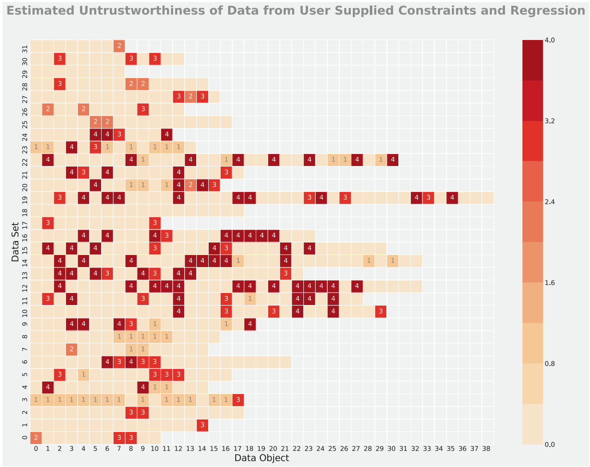


Fig. 4: Class in-specific quarantine results for all data showing the sum of all indicators of suspected low-integrity for user-defined constraints and `food_info` regression constraints. Higher numbers indicate a higher estimation of low-integrity. Where no numbers are given, none of the data-integrity constraints indicated low-integrity. Data set three has indications of low integrity on the manufacturers data, and all points matching the manufacturers data due to the data supplied by the manufacturer being out of compliance with FDA labeling regulations.

References

[1] P. Zikopoulos, K. Parasuraman, T. Deutsch, J. Giles, D. Corrigan *et al.*, *Harness the power of big data The IBM big data platform*. McGraw Hill Professional, 2012.

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, “Basic concepts and taxonomy of dependable and secure computing,” *IEEE transactions on dependable and secure computing*, vol. 1, no. 1, pp. 11–33, 2004.

[3] H. Gao, G. Barbier, and R. Goolsby, “Harnessing the crowdsourcing power of social media for disaster relief,” *IEEE Intelligent Systems*, vol. 26, no. 3, pp. 10–14, 2011.

[4] M. Jackson, H. Rahemtulla, and J. Morley, “The synergistic use of authenticated and crowd-sourced data for emergency response,” in *2nd International Workshop on Validation of Geo-Information Products for Crisis Management (VALgEO)*, 2010, pp. 91–99.

[5] M. Swan, “Health 2050: the realization of personalized medicine through crowdsourcing, the quantified self, and the participatory biocitizen,” *Journal of personalized medicine*, vol. 2, no. 3, pp. 93–118, 2012.

[6] ——, “Crowdsourced health research studies: an important emerging complement to clinical trials in the public health research ecosystem,” *Journal of medical Internet research*, vol. 14, no. 2, 2012.

[7] Y. Demchenko, P. Grosso, C. De Laat, and P. Membrey, “Addressing big data issues in scientific data infrastructure,” in *Collaboration Technologies and Systems (CTS), 2013 International Conference on*. IEEE, 2013, pp. 48–55.

[8] C. Cachin, J. Camenisch, M. Dacier, Y. Deswarte, J. Dobson, D. Horne, K. Kursawe, J. Laprie, J. Lebraud, D. Long *et al.*, “Malicious-and accidental-fault tolerance in internet applications: reference model and use cases,” LAAS report, Tech. Rep., 2000.

[9] R. Grigonis, “Fault-resilience for communications convergence,” *Special Supplement to CMP Media’s Converging Comm. Group, Spring*, 2001.

[10] K.-K. Muniswamy-Reddy and M. Seltzer, “Provenance as first class cloud data,” *ACM SIGOPS Operating Systems Review*, vol. 43, no. 4, pp. 11–16, 2010.

[11] E. Rahm and H. H. Do, “Data cleaning: Problems and current approaches,” *IEEE Data Eng. Bull.*, vol. 23, no. 4, pp. 3–13, 2000.

[12] I. Guyon, N. Matic, V. Vapnik *et al.*, “Discovering informative patterns and data cleaning.” 1996.

[13] D. L. Parnas, “On the criteria to be used in decomposing systems into modules,” *Communications of the ACM*, vol. 15, no. 12, pp. 1053–1058, 1972.

[14] F. Cristian, “Understanding fault-tolerant distributed systems,” *Communications of the ACM*, vol. 34, no. 2, pp. 56–78, 1991.

[15] P. K. Infrastructure and T. P. Profile, “Common criteria for information technology security evaluation,” *National Security Agency*, 2002.

[16] Y. Demchenko, C. de Laat, and V. Ciaschini, “Vo-based dynamic security associations in collaborative grid environment,” in *Collaborative Technologies and Systems, 2006. CTS 2006. International Symposium on*. IEEE, 2006, pp. 38–47.

[17] Y. Demchenko, A. Wan, M. Cristea, and C. De Laat, “Authorisation infrastructure for on-demand network resource provisioning,” in *Proceedings of the 2008 9th IEEE/ACM International Conference on Grid Computing*. IEEE Computer Society, 2008, pp. 95–103.

[18] Y. Demchenko, C. de Laat, L. Gommans, B. Oudenaarde, A. Tokmakoff, and M. Snijders, “Job-centric security model for open collaborative environment,” in *Collaborative Technologies and Systems, 2005. Proceedings of the 2005 International Symposium on*. IEEE, 2005, pp. 69–77.

[19] Y. Demchenko, M. Cristea, and C. De Laat, “Xacml policy profile for multidomain network resource provisioning and supporting authorisation infrastructure,” in *Policies for Distributed Systems and Networks, 2009. POLICY 2009. IEEE International Symposium on*. IEEE, 2009, pp. 98–101.

[20] S. Kripke, “Semantical considerations of the modal logic,” 2007.

[21] R. R. Stoll, “Sets, logic, and axiomatic theories,” 1961.

[22] P. R. Halmos, *Naive set theory*. Courier Dover Publications, 2017.

[23] M. Abadi and L. Cardelli, “An imperative object calculus,” *TAPSOFT’95: Theory and Practice of Software Development*, pp. 469–485, 1995.

[24] ——, “A theory of primitive objects: Untyped and first-order systems,” *Information and Computation*, vol. 125, no. 2, pp. 78–102, 1996.

[25] L. Cardelli, “Type systems,” *ACM Computing Surveys*, vol. 28, no. 1, pp. 263–264, 1996.

[26] M. A. AbdelGawad, “Noop: A mathematical model of object-oriented programming,” Ph.D. dissertation, Rice University, 2012.

[27] US Food and Drug Administration and others, “Guidance for industry: a food labeling guide,” *Food and Drug Administration, Washington, DC, USA*, 2008.

[28] G. McLachlan, K.-A. Do, and C. Ambroise, *Analyzing microarray gene expression data*. John Wiley & Sons, 2005, vol. 422.

[29] B. E. Boser, I. M. Guyon, and V. N. Vapnik, “A training algorithm for optimal margin classifiers,” in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.

[30] A. Liaw, M. Wiener *et al.*, “Classification and regression by randomforest,” *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[31] M. Dezani-Ciancaglini, P. Giannini, and E. Zucca, “Extending the lambda-calculus with unbind and rebind,” *RAIRO-Theoretical Informatics and Applications*, vol. 45, no. 1, pp. 143–162, 2011.

[32] V. P. Nelson, “Fault-tolerant computing: Fundamental concepts,” *Computer*, vol. 23, no. 7, pp. 19–25, 1990.

[33] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi, “A taxonomy of computer program security flaws,” *ACM Computing Surveys (CSUR)*, vol. 26, no. 3, pp. 211–254, 1994.

[34] J.-C. Laprie, “Dependable computing and fault-tolerance,” *Digest of Papers FTCS-15*, pp. 2–11, 1985.

[35] D. Powell, R. Stroud *et al.*, “Conceptual model and architecture of maftia,” *Technical Report Series-University of Newcastle Upon Tyne Computing Science*, 2003.

[36] C. E. Brodley and M. A. Friedl, “Identifying mislabeled training data,” *Journal of artificial intelligence research*, vol. 11, pp. 131–167, 1999.

[37] D. Crockford, “The application/json media type for javascript object notation (json),” 2006.

[38] G. Van Rossum and F. L. Drake Jr, “The python language reference,” *Python software foundation*, 2014.

[39] L. MyFitnessPal, “Myfitnesspal,” 2015.

Natural Language Processing based Optimization Methods

Reducing Operator Overload with Context-Sensitive Event Clustering

Marcus Basalla, Johannes Schneider and Jan vom Brocke
University of Liechtenstein, 9490 Vaduz, Liechtenstein
{marcus.basalla, johannes.schneider, jan.vom.brocke}@uni.li

***Abstract*— Operators of complex, networked systems are constantly confronted with a large number of error events that are time-consuming to address. Events in one network component can trigger a cascade of events in other components leading to many intertwined sequences of a large number of error messages. Operators often only seek to identify and understand the root cause of a sequence of events indicating a problem, since addressing the root cause commonly elevates consequential issues. Based on a real world dataset we introduce two techniques to reduce the number of events and error logs without discarding the root cause. One technique leverages existing process mining tools combined with manual analysis. The other relies on computing context sensitive embeddings, similar to word embeddings in natural language processing. The embeddings are used to cluster event types to identify co-occurrence and causality between them. While both techniques have their strengths and weaknesses, they each significantly reduce the number of possible events, while enforcing conditions for causality.**

Keywords— Event Logs, Error Logs, Root cause identification, Word2Vec, Clustering, Process Mining.

I. Introduction

In complex, networked systems error events occur almost constantly. Within milliseconds an error in one machine can trigger an error in another machine and so on. Thus, interrelated errors are accumulating quickly. While at the same time, other independent errors might occur. The sheer amount of error messages confronting a human operator can easily lead to the problem of information overload [1]. The machine operators can be overwhelmed with thousands of error messages that make it difficult and time-consuming to figure out the exact nature of the problem, i.e., the root cause. The root cause is the initial error event that triggered all the consecutive error events. Such initial events might occur almost simultaneously at different network entities leading to multiple intertwined event sequences that might potentially also interact. This is illustrated in Figure 1. We assume that our event log contains no structuring, i.e. there is no obvious partitioning into traces, but rather a stream of more or less continuous events.

In this work, we aim to transform error logs by reducing the number of errors displayed to an operator who is tasked with root cause identification. This is achieved by removing or grouping error events. Our goal is not to fully automatically identify all root causes, but rather to reduce the workload of the operator by reducing the event log size. Once a root cause is identified in the reduced event log, an operator might validate the finding in the original log. As we do not focus on perfectly identifying root causes themselves but on reducing the size of the event log without discarding root causes, we can make use of co-occurrences and their order as a proxy for causal effects between event types.

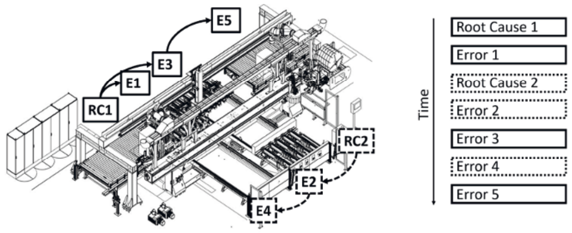


Fig. 1 Example of an error log (right side) triggered by two independent root cause events occurring at different locations in the production site (left side).

We apply two techniques: (standard) process discovery [2, 3] combined with manual analysis and a method inspired by word embeddings combined with clustering. For each cluster (independently), rules to remove or merge event types are applied. Both allow to significantly reduce the number of candidate events for root causes by enforcing logical conditions regarding causality. However, relaxing (strict) causality conditions is generally beneficial. That is, allowing for a slight risk of removing a root cause or a somewhat larger risk of falsely reporting a root cause, can re-duce event logs significantly more. We apply both techniques to a real-world dataset from a production site of a company that offers batch size systems for state-of-the-art fully automated, networked, and digitized production. We achieve a reduction of about a factor of 5 with fairly strict causality constraints.

II. Related Work

Most work on event logs focuses on root cause identification based on graph models [3, 4]. Several recent papers focus on discovering and utilizing causal relationships to this end [5, 6, 7]. For example, [5] generates a partial ancestral graph from a situation feature table and uses causal structure learning to generate a graph of causal relationships. However, these works rely on enriching the event log with additional information e.g. a business process model or known information about the process each entity of the log occurs in. In contrast, our methods focus on the situation where such information cannot be easily accessed. Further, these works differ from ours as they aim for a clear identification of causal structures, while we focus solely on reducing the complexity of the event logs without missing root causes. In this regard, our work is similar to [8], which uses co-occurrence and causal rules to detect anomalous process traces.

Similar to [9] our work uses a clustering approach to identify clusters of event logs. However, they rely on human experts to further annotate these clusters and event descriptions, while our

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_9

method relies only on collected data, i.e., the context in which an event occurs in the event log. Another field related to root cause identification is the detection of anomalies in complex event logs [10, 11]. For example, [11] applies a Word2Vector based event log representation similar to the one used in our clustering solution. However, instead of event log reduction, they focus on the detection of anomalous event sequences. This method is similar to trace embeddings, where instead of individual events, entire event traces are encoded as vectors [12]. The seminal work introducing the α-algorithm [3] is for a different setup, where a workflow log is logically organized into cases. Identifying the root cause given a "case", i.e. a sequence of events, is trivial: It is simply the first event. However, if the "case IDs" are missing, the problem becomes more difficult.

III. Methodology

The overall strategy is to identify relationships between event types that allow to reduce the number of potential root causes by either excluding event types or combining multiple event types. Before describing our two techniques, we first formally define two core concepts characterizing the event log and problem.

Definition 1a: An event log $E = (e_1, e_2, \ldots, e_n)$ is a sequence of events $e_k = (c_k, t_k)$, with c_k being an event type (or class), and t_k being the time the event was registered. To allow interpretation of the error by human operators, each class c_k is associated with an event message m_k.

Definition 1b: The set of events et_c of a specific event type c is the subset of all events in E of the same type c:

$$et_k = \{e_i = (c_i, t_i) \mid c_i = c\}$$

We used two approaches: For the first, we compute event type embeddings based on all events of a specific type and the context in which they occur. Then, we cluster event types. This allows us to reduce the number of comparisons for each event type to assess interdependencies to the most likely candidates. More precisely, we apply Word2Vec [10] to calculate an embedding for each event type. We use the agglomerative hierarchical clustering [14] to group event types into clusters. As Word2Vec embeddings are context sensitive, meaning words that appear in a similar context in the text have similar embeddings, the idea is that event types in one group typically bear some relationship, i.e. are commonly found in temporal proximity, while event types in different clusters are more likely unrelated and occur relatively rarely within a short time span. Then, we compute a co-occurrence matrix between event types for each cluster. Finally, we check which pairs of events within a cluster are root causes and which event types can be merged or discarded. An illustration of the methodology is shown in Figure 2. The second methodology is based on process mining using the process discovery software "Disco"[1].

A. *EventType2Vec*

EventType2Vec assigns a vector to each event type as is done for words in the context of text mining, i.e. Word2Vec [13]. Word2Vec is a technique from natural language processing for statistical analysis of texts. Word2Vec has been applied in prior work in the context of log analysis [11]. It is based on the idea to represent words as vectors. A vector is sensitive to the context in which a word typically appears (i.e. the words before and after) and condenses this information into a vector by using a small, shallow neural network [13]. Thus, Word2Vec might be seen as a dimensionality reduction of the co-occurrence matrix that results from partitioning texts into small segments, counting pairs of words within each segment, and aggregating the counts. The widespread success of Word2Vec is arguably due to two factors, namely, its computational speed, allowing it to process a large amount of data, and its novel computational properties. It allows computing similarities between words (rather than just comparisons for equality), which is useful for many applications. Vectors also allow for arithmetic operations, e.g. the formula "king – man + woman" yields roughly the vector for "queen". Such operations are not easily possible when words are represented as tokens or as one-hot encodings, i.e., a vector with just a single entry differing from zero.

When using Word2Vec for error events, we interpret the sequence of events as one large text document and the instances of events as tokens. Each token is of one specific event type. EventType2Vec vectors for each event type are computed using subsequences of the event log of a fixed length. The subsequence length is defined by a window size parameter w. For each event, all $w/2$ previous and all $w/2$ consecutive events are considered as the context. The choice of the parameter w depends heavily on the expected overlap of sequences of events of independent root causes. The more sequences overlap the larger w should be. For illustration, consider a common sequence of events (e_1, e_k) having root cause e_1 and a second event e_k. If a lot of independent errors occur, the two events might always be separated by multiple other events, e.g. in the log an instance might appear as $(e_1, e_2, e_3, \ldots, e_k)$. Thus, if a small window size is used e_1, e_k do not occur in the same window and, thus, they will be treated as unrelated. However, w should not be chosen too large, i.e., if it is chosen two be the entire size of the event log, all contexts are the same for all event types. That is, too large w fosters similarity between unrelated event types.

The similarity score based on vectors of event types ranges from -1 to 1. A value of "1" denotes maximal positive correlation, i.e., the two event types always occur together. A value of "-1" shows a maximum negative correlation, i.e. the presence of one event type in a specific part of the sequence coincides with the absence of the other. The property of Word2Vec that makes it interesting for this task is that event types, which usually occur in the same context, also have resembling similarity vectors [13].

[1] https://fluxicon.com/disco/

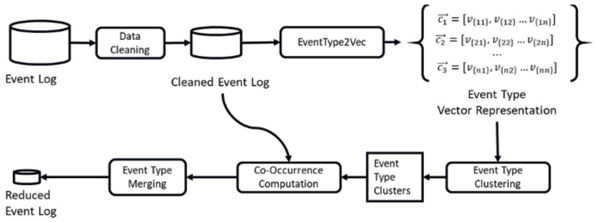

Event Log
Data Cleaning
Cleaned Event Log
EventType2Vec
$\vec{c_1} = [v_{\{11\}}, v_{\{12\}} \dots v_{\{1n\}}]$
$\vec{c_2} = [v_{\{21\}}, v_{\{22\}} \dots v_{\{2n\}}]$
…
$\vec{c_3} = [v_{\{n1\}}, v_{\{n2\}} \dots v_{\{nn\}}]$
Event Type Vector Representation
Event Type Clustering
Event Type Clusters
Co-Occurrence Computation
Event Type Merging
Reduced Event Log

Fig. 2 Overview of the EventType2Vec based clustering approach

B. Agglomerative Clustering

Agglomerative hierarchical cluster analysis (AHCA) allows the creation of clusters from unlabeled data based on how relatively similar/dissimilar they are [14]. That is, event types in the same group (or cluster) are similar and require manual analysis, while event types in distinct clusters are dissimilar and do not need to be investigated. This allows us to reduce the number of event type combinations that have to be considered as possible co-occurring, i.e. bearing any form of relationship. The maximum reduction is from a single cluster with $n^2/2$ combinations down to k clusters each with $n^2/(2k^2)$ combinations to consider. Thus, the number of clusters k controls a trade-off between the time that the search for co-occurring pairs takes by the operator and its recall. Increasing the number of clusters speeds up computation, but increases the likelihood of a pair being neglected since we treat clusters in isolation, i.e. we assume that if an event type A is in one cluster and type B in another then they do not bear any relationship.

The clustering objective of AHCA is to minimize the sum of distances within every cluster. AHCA does not take into account if clusters strongly differ concerning the number of components, which means that a single event type can end up in a cluster on its own. Thus, the effective benefit of using k clusters is highly data-dependent. A single event type in a cluster indicates that this event type bears little to no relation to any other type of event and hence can be ignored in further analysis. On the other hand, events that occur in similar contexts have a small distance between their vector representations. Therefore, they are likely to appear in the same cluster.

To quantify how 'relatively similar' objects are, we used the Euclidean distance. We applied standard AHCA, where distances between all vectors are computed. Since it requires O(n^2) run-time, computational speed is a concern for large n, i.e., a large number of event types. In this case, one might resort to approximation algorithms [15, 16] that might rely on some randomness in contrast to classical methods that allow for better reproducibility.

C. Co-Occurrence Detection

Error events usually propagate within a specific time span. Thus, if two events have a large time difference, they can be said to be unrelated. More precisely, within our log, we say that two events e_1 and e_2 are unrelated, if the time difference in their occurrence is more than a threshold s, i.e. $|\, t_1 - t_2| > s$. Based on this threshold we can define a value that measures the co-occurrence rate of two event types as follows:

Definition 2: The co-occurrence rate of two events types within the time frame of s seconds is defined as

$$co(c_1, c_2) = \frac{\left|\{(e_i, e_j) | e_i \in et_{c1}, e_j \in et_{c2}, |t_1 - t_2| \le s, \}\right|}{|et_{c1}|}$$

The co-occurrence rate lies within the range of [0,1], where 0 signifies no co-occurrence and 1 signifies that always either events of both types or none of the two types occur with the time frame of s. It is important to note that the rate is not symmetric, i.e., generally, $co(c_1, c_2) \neq co(c_2, c_1)$. Equality holds if both have the same count, i.e. $|\{et_1\}| = |\{et_2\}|$. This is also intuitive: If events of one type occurs much more frequently than that events of another type then the co-occurrence of the more frequent event type should be lower since it cannot always occur together with the rarer event type.

The definition of the time threshold s might also be event type specific, however, in this work, we focus on using a global threshold, since, for our case study, events tend to occur consistently within fairly short time spans. Choosing the threshold too large, yields many spurious, non-relevant relationships, i.e. events are associated that originate from different root causes. It yields a lower reduction of event logs. Choosing it too small, tends to yield more root causes than there actually are. If the time between two events exceeds the (small) threshold, then a single chain of events (with a single root cause) might be treated as multiple chains of events (each with an alleged separate root cause).

To identify the temporal relationships between event types within the same cluster, we compute a co-occurrence matrix based on Definition 2. As the co-occurrence matrix is calculated separately for each of the k (small) clusters, the method is much more efficient than relying on one large cluster, i.e., $n^2/2$ down to $n^2/(2k^2)$. To identify potential event types that can be merged or deleted, we apply a threshold t on the co-occurrence rate. For this, we introduce the following notation that essentially demands that events of one type always precede that of another type.

Definition 3: We denote co-occurrences above the threshold f with $c_1 \overset{f}{\widetilde{\approx>}} c_2$ if $co(c_1, c_2) \ge f$. We will use the simplified notation $c_1 \approx> c_2$ for a fixed threshold f selected by the operator.

Based on this definition we set forth the following rules to simplify our error logs.

Rule 1: if $c_1 \approx> c_2$ there is either a causal relation between c_1 and c_2 or they share the same root cause.

Therefore, by merging the two event types we do not delete the root cause but preserve it earlier in the log or in the merged event type.

Rule 2: if $c_1 \approx> c_2$ and $\forall \left(e_i = (c_1, t_i).\, e_j = (c_2, t_j)\right) : t_1 < t_2$then either c_1 causes c_2 or they share the same root cause.

TABLE I. EVENT PATTERNS AND THEIR HANDLING TO REDUCE THE NUMBER OF ENTRIES.

Pattern	Description	Transforming Strategy
Recurrence of a single event type	Events of the same type occur many times in a short time span. No significant co-occurrence with other events.	Keep only once
Undirected pairs	A pair of events that show strong undirected co-occurrence (A≈>B) and (B≈>A)	Combine (A, B) into one new event G occurring whenever event A would occur.
Directed Pairs	Pair of events that show strong co-occurrence in one direction (A≈>B) but not the other (B≈>A)	Combine (A, B) into one new event group G whenever A occurs; Occurrences of B without A remain unchanged

Therefore, by removing an event of type c_2 whenever an event of type c_1 occurs, we do not delete the actual root cause from the log. However, we can wrongly declare c_1 as the root cause, when the actual root cause occurs earlier in the event log and it is not treated as co-occurring, i.e. it is below the threshold t. Furthermore, we can also declare another event type c_3 wrongfully as a root cause, if $c_1 \approx> c_2$ and $c_2 \approx> c_3$. The latter can be easily avoided if we only remove an event type c_2 if $c_1 \approx> c_2$ and there exists no c_3 such that $c_2 \approx> c_3$.

For a threshold of f = 1 (100% co-occurrence) the above rules do not allow that a root cause event gets removed. However, if the threshold f is very low (significantly below 0.5) and events originating from different root causes are intertwined in certain ways, root cause events might be removed. However, keeping the threshold at f = 1, is not favorable, since if the order of events is identical in all, possibly millions of cases, but differs once, the event types are treated as not having a causal relationship. For illustration, consider two common sequences of events with types (c_1, c_2) and (c_3, c_2) having root causes c_1 and c_3. If in the event log only once there is a sequence being the concatenation of the two $((c_3, 1), (c_2, 2), (c_1, 3), (c_2, 4))$ then no causal relationship between c_1 and c_2 will be assumed.

An initial analysis of a sub-sample of the dataset leads to the discovery of different patterns of event pairs analogous as for the α-algorithm [3] or any other algorithm aiming to fulfill causality constraints, where precedence of events is listed as one condition. Table 1 summarizes these patterns and the strategy that can be applied to reduce the number of event logs. Some pairs show an almost perfect co-occurrence, where error A almost always predicts error B (A≈>B) and the other way around (B≈>A). These are easiest to handle in terms of reducing error messages as each pair can be combined into a new (aggregated) event group G. That is, each pair of co-occurring events (A, B) or (B, A) in the log is replaced by a new event "type" G replacing the two events. Note that our definition of co-occurring allows for a time gap of s between two events, i.e. they might not appear directly after each other, i.e. we replace (A, C, D, B) with (G, C, D).

Relations do not necessarily occur only between pairs. Larger clusters of co-occurrences are possible, e.g. for elements A, B, and C we could have (A≈>B), (B≈>A), (A≈>C), (C≈>A), (B≈>C), and (C≈>B). Thus, multiple event types can be combined into a single event type group. Technically, these cases can be covered by iteratively applying the merging rule for pairs (taking into consideration newly merged event pairs) until no further reduction is possible, e.g. for (A≈>B), (B≈>A), (A≈>C), (C≈>A), (B≈>C), and (C≈>B), we merge (A, B) into G yielding (G≈>C), (C≈>G). Then, we merge (G, C) into G'.

An event type can also form a cluster on its own, i.e. the cluster consists only of the single event type. Such an event type does not show significant similarity with any other event type. If event messages of this type occur regularly within the event log and within a short time of other events of the same type, this is an indicator for redundant messages. A possible cause can be a problem that generates an error message continuously in a fixed interval until it is solved.

D. Process Mining

The second approach is based on process mining and more specifically the analysis of process maps. This approach used the process mining software "Disco" for process discovery. The software was used repeatedly to refine clusters, excluding them from the dataset, and creating a new process map from the remaining events.

To identify different kinds of event clusters a manual strategy was applied. This strategy was developed based on the CRISP-DM methodology and an initial manual inspection of a subset of the dataset. Figure 3 shows an overview of this process. After cleaning the event log a process map is generated. Figure 4 (left) shows an excerpt of the initial process map. The operator manually identifies continuous paths of events, i.e. they appear as chained events. Figure 4 (right) shows an example of such a path found in the dataset. If the initial event of a path does not have any more dependencies it can be identified as a root cause. All events in a path after a root cause are deleted as we assume that resolving the root cause error resolves all depending errors. If the first event of the path is no root cause, all events on the path are merged into one event that contains the event messages of all merged events. While this does not directly identify a root cause, it still allows to reduce the number of events in the log and therefore the overall number of events. This process is formalized in Algorithm 1.

Disco requires a process ID to identify separate events to properly generate a process map. As this is not explicitly defined in the original dataset, the hostname has been used to separate processes. As a result of this decision, only dependencies within one hostname are found through the process mining approach.

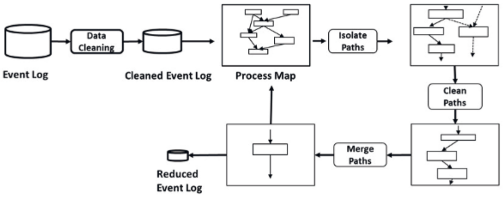


Fig. 3 Overview of the Process Mining approach.

Algorithm 1: Process Mining Work Flow for Event Log Reduction

```
Visualize the error log E as a process map.
Identify and exclude event types without dependencies and duplicates.
Find paths of at least two event types: $p_i = c_1 -> c_2 -> \cdots -> c_n$
for each $p_i$:
    if $c_1$ has no dependencies:
        Mark $c_1$ as root cause
    else:
        Replace $c_1$ with new event type $c_{p_i}$
    Delete $c_2,\ldots,c_n$
```

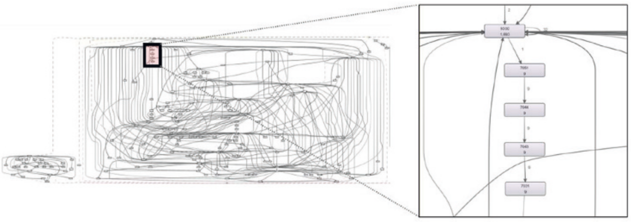


Fig. 4 Left: Excerpt from the process map generated by the tool “Disco” in the first iteration of the process mining approach. Line strength shows association strength between events. Right: Example of a potential cluster given by a path. There is a clear sequence of event dependencies without any forking.

IV. Evaluation and Results

A. Dataset and Preprocessing

The original dataset had 3,563,724 events in one large log file with 503 different error event types and 18 of them occurring more than 10,000 times. Error events are potentially logged repeatedly until the error is solved. We removed duplicates by keeping only the first event for a subsequence of events of the same type. One example of this is the error “FMC file not found”, which corresponds to a share of 76% of the entire event log. This message appears thousands of times in a row, with each message only separated by at most one second. Since this error originates from a remote client, an overload could be avoided by reducing the notification interval or even changing the notification procedure in such a way that this client sends the information about the missing file just once and waits until the problem is handled. By applying these adaptations to the production setup, the event log reductions from pre-processing can be carried over to an online application. Overall, preprocessing reduced the dataset to 732,579 event logs without excluding any event type from the dataset.

B. *Clustering*

Before testing our method on the complete dataset, we first applied it to a small subset of the event log, in order to identify a good configuration for the model parameters. For the Word2Vec encoding, we chose a window size w of 100 events. There is no strong sensitivity to the parameter w, i.e. w of size 50 and 200 yield comparable results. For the agglomerative clustering, we set the number of clusters to k=50 and used an Euclidean distance measure. Thus, given that we have 503 event types, this yields an average cluster size of 10 event types. Since clusters are analyzed manually, we found an average size of 10 to be a well-manageable size. Otherwise, we relied on the standard parameters provided by the python library scikit learn. For computing the co-occurrence value, the co-occurrence time window s from Definition 2 was chosen as s = 60 seconds. The threshold for considering a co-occurrence from Definition 3 was set to f = 0.85.

Next, we present some of our findings for the specific use case. We discussed these and other findings with a domain expert from our industrial partner, which deemed them reasonable.

One interesting outcome of using EventType2Vec is that events that appear very often and without any apparent relationship to any other type of event also have highly unique similarity vectors, meaning each one of these is put into a cluster on its own and hence does not need to be checked any further. An extreme example of this is the event (“FMC file not found”).

Following the clustering and co-occurrence computation described in Sections 4.1 to 4.3, a total of 3 event pairs were detected. An example for an undirected pair are the event types with the ids 54X05 and 54X06. Combining these two events into one reduces the number of messages per day by around 100.

An undirected cluster is formed by four event types that occur around 600 times in the dataset and their pairwise co-occurrences range between 92% and 100%. The root cause seems to be a problem with the supply voltage, which in most cases precedes the other three messages, whose temporal order follows varying patterns, though having only median distances of a few seconds. Merging these messages helps decrease the total size of the event log by around 1800 events.

There also seems to be some link between the event pair 61/62 and event 9300, which occurs in 87.5% of cases together with or a few seconds before event 61. If this happens, the three errors could be condensed into one message. However, there are also many cases where 9300 occurs on its own, making it an example of a directed pair. So no general merging of 61/62 and 9300 should be performed.

C. *Process Mining*

A specific process in our dataset concerned the handling of external files. Each step in the file processing generated a different message (e.g., start, run, done, error, file not found) which made up a quarter of the analyzed dataset. For this process, only the outcomes are relevant which means only the logs “Done”, “Error”, or “File not found” are relevant and, ideally, “Error” and “Done” should only occur once for each file. Like in the EventType2Vec analysis, this process generated many redundant error events. Handling this specific process helped reduce the original 188,602 logs concerning files to only 1,629 logs. Overall, process mining used 710,852 different error occurrences, which were reduced to 303,692 error occurrences through getting rid of unnecessary events and finding duplicates. Furthermore, 30 different processes could be found within these 303,692 events, while 161,045 events out of these do not seem to have dependencies, the remaining 142,647 logs could be further reduced to 55,548 by applying the strategies summarized in Table 2. In the end, the initial 710,852 initial event logs could be reduced to 216,593.

V. Discussion and Limitations

While both methods show a similar reduction of events, they both have their advantages and disadvantages. The largest downside of the process mining methodology is its reliance on a manual operator in each iteration. The application of EventType2Vec followed by clustering on the other hand relies heavily on cluster parameters both in the resulting quality and the run time. This is a considerable obstacle to usability in a real world context, where workers are generally not familiar with such highly technical parameters. Still, in principle, the computation of an embedding and the clustering can be skipped, if the problem size is small or large computational resources are available. Further, in contrast to the process mining approach, using EventType2Vec clustering and the following simplification rules does not incorporate any knowledge about the dataset. On the one hand, this is an advantage as it highlights the generality of the method. It can be used for other datasets from unrelated fields and it is applicable in cases where there is no expert available. On the other hand, there is also no possibility for an expert to incorporate their knowledge into the algorithm. In practice, a combination of both approaches will likely lead to the best results. One could for example imagine a process in which clustering is applied first to identify the most common event type co-occurrences and redundancies to reduce the workload of a more manual process mining approach performed by an expert further down the line.

Our research focuses on the reduction of event logs in complex networked systems. As this scenario does still rely on a human operator to further analyze the reduced logs, it is not a fully automatic method for identifying error causes. We discussed the outcomes with our industrial partner. While the partner was pleased, a more exhaustive qualitative component based on the evaluation of actual operators would be desirable. However, the decrease in the total amount of event logs, along with the fact, that based on the co-occurrence measures from Definition 3 no root causes can be deleted, there is a clear case for practical relevance of the methods. This includes functionality to visually explore a reduced event log and drill down to expand subsequences of merged events to their original non-merged counterparts. A future study should focus on quantitatively evaluating the method on a dataset with labeled ground truths for event relations and root causes. Our methodology primarily helps in pointing towards potential root causes but the final verification of them is still a manual approach. An indicator quantifying the likelihood that an identified root cause is actually a root cause might be helpful.

VI. Conclusion

We introduced and evaluated two methods for reducing the number of events in error logs presented to human operators. This helps to reduce information overload and facilitate faster problem identification. The process mining approach relies more heavily on a human operator but allows the incorporation of domain knowledge, while the clustering methodology based on vectors from EventType2Vec requires no knowledge of the semantics of events but also does not allow the direct incorporation of such knowledge to improve the quality of the outcome. While both methods achieve a considerable event log reduction on their own, a combination of both methods, where the more automated Word2Vec clustering is used to reduce the event logs for more manual process mining seems to be promising. In future work, we also aim to leverage other NLP techniques, e.g. topic modeling to discern event traces and identify root causes [17], potentially in combination with explainability methods.

Acknowledgements

We thank our industrial partner and all people working on the project, in particular, Claudio Gall and Schäfer Mathias.

References

[1] Schick, A. G., Gordon, L. A., & Haka, S.: Information overload: A temporal approach." Accounting, Organizations and Society, 15(3), 199-220, 1990.

[2] Van Der Aalst, W.: Data science in action. Springer, 2016.

[3] Van der Aalst, Wil, Ton Weijters, and Laura Maruster. "Workflow mining: Discovering process models from event logs." IEEE transactions on knowledge and data engineering, 2004.

[4] de Leoni, M., van der Aalst, W.M., Dees, M.: "A general process mining framework for correlating, predicting and clustering dynamic behavior based on event logs." Inf. Syst., 2016.

[5] Qafari, M. S., & van der Aalst, W.: "Root cause analysis in process mining using structural equation models". International Conference on Business Process Management, 2020.

[6] Hompes, B. F., Maaradji, A., La Rosa, M., Dumas, M., Buijs, J. C., & van der Aalst, W. M. : "Discovering causal factors explaining business process performance variation." In International Conference on Advanced Information Systems Engineering pp. 177-192, 2017.

[7] Narendra, T., Agarwal, P., Gupta, M., & Dechu, S.: "Counterfactual reasoning for process optimization using structural causal models." International Conference on Business Process Management, 2019.

[8] De Koninck, Pieter, Seppe vanden Broucke, and Jochen De Weerdt. "act2vec, trace2vec, log2vec, and model2vec: Representation Learning for Business Processes." International Conference on Business Process Management, 2018.

[9] Cotroneo, D., Paudice, A., & Pecchia, A.: "Automated root cause identification of security alerts: Evaluation in a SaaS Cloud." Future Generation Computer Systems, 2016.

[10] Gupta, N., Anand, K., Sureka, A.: Pariket: "mining business process logs for root cause analysis of anomalous incidents." DNIS, 2015.

[11] Wang, J., Tang, Y., He, S., Zhao, C., Sharma, P. K., Alfarraj, O., & Tolba, A., "LogEvent2vec: LogEvent-to-vector based anomaly detection for large-scale logs in internet of things." Sensors, vol. 20(9), 2451 (2020).

[12] Weidlich, M., Ziekow, H., Mendling, J., Günther, O., Weske, M., & Desai, N., "Event-based monitoring of process execution violations." International conference on business process management, 2011.

[13] Mikolov, T., Chen, K., Corrado, G., & Dean, "J. Efficient estimation of word representations in vector space." arXiv:1301.3781, 2013.

[14] Aljumily, R.: "Agglomerative hierarchical clustering: An introduction to essentials." Global Journal of Human-Social Science, 2016.

[15] Schneider, M Vlachos, "On randomly projected hierarchical clustering with guarantees." In Proceedings of the 2014 SIAM International Conference on Data Mining, pp. 407-41, 2014.

[16] Schneider, J., & Vlachos, M., . "Scalable density-based clustering with quality guarantees using random projections." Data Mining and Knowledge Discovery, vol. 31(4), pp. 972-1005, 2017.

[17] Schneider, Johannes, and Michail Vlachos. "Topic modeling based on keywords and context." Proceedings of the 2018 SIAM international conference on data mining. Society for Industrial and Applied Mathematics, 2018

[18] Meske C, Bunde E, Schneider J, Gersch M. "Explainable artificial ingelligence: objectives, stakeholders, and future opportunities." Information Systems Management, 2021

Dynamic Review-based Recommenders

Kostadin Cvejoski[*‡], Ramsés J. Sánchez[*†], Christian Bauckhage[‡] and César Ojeda[§]
[*]Competence Center Machine Learning Rhine-Ruhr, 53757 Sankt Augustin, Germany
[†]University of Bonn, 53113 Bonn, Germany
[‡]Fraunhofer Center for Machine Learning and Fraunhofer IAIS, 53757 Sankt Augustin, Germany
[§]Berlin Center for Machine Learning and TU Berlin, 10587 Berlin, Germany
{kostadin.cvejoski, christian.bauckhage}@iais.fraunhofer.de,
ojeda.marin@tu-berlin.de, sanchez@bit.uni-bonn.de

***Abstract*—Just as user preferences change with time, item reviews also reflect those same preference changes. In a nutshell, if one is to sequentially incorporate review content knowledge into recommender systems, one is naturally led to dynamical models of text. In the present work we leverage the known power of reviews to enhance rating predictions in a way that (i) respects the causality of review generation and (ii) includes, in a bidirectional fashion, the ability of ratings to inform language review models and vice-versa, language representations that help predict ratings end-to-end. Moreover, our representations are time-interval aware and thus yield a continuous-time representation of the dynamics. We provide experiments on real-world datasets and show that our methodology is able to outperform several state-of-the-art models. Source code for all models can be found at [1].**

***Index Terms*—recurrent recommender networks, dynamic language model, attention for recommendation**

I. Introduction

Following the deep learning agenda, the success of modern recommender systems heavily relies on their ability to leverage meaningful representations that allow for the accurate prediction of a purchase or a rating. Fundamentally one must dwell into *interest modeling*, as an effective recommendation is such that it uncovers, for a given user, a hidden interest in an unknown item. It is natural to study the change of user interest with time and, in the present work, we seek to incorporate these dynamical notions with those of text reviews.

Reviews are effectively a form of recommendation, and one that is directly provided by the user. The challenge however, stems from the unstructured and ambiguous nature of reviews (and natural language itself). A user might, for example, simultaneously highlight positive and negative aspects of the different items she reviews. Following current trends in natural language processing, we leverage review content through neural models of text and attention mechanisms, and guarantee the information content of those representations via reproduction quality. We encourage the dynamical aspect of text recommendations by learning representations which help predict both *when* is the next review arriving and *what* does it say. One is then led naturally to dynamical language models, since enforcing good text predictions ensures its dynamical representation quality.

II. Related Work

There is a large body of research invested in recommender systems (RS), a big part of which has lately been devoted to capture the temporal dynamics of both users and items. One of the first temporal models for recommendation is the TimeSVD++ [2], which extends the SVD++ matrix factorization algorithm by introducing time-dependent latent factors. From the neural network perspective, many models for RS have been developed [3]–[5], and Recurrent Neural Networks (RNNs) have been used to capture time-ordered user activity. For example, session-based item recommendation use RNNs to infers user preferences from sessions of user behaviour [6]–[9]. Another example, closer to our work, is the Recurrent Recommender Networks (RRN) which uses two independent RNNs to model user and item dynamics separately [10].

Just as with user (and item) temporal representations, including review content representations has also been shown to significantly improve rating prediction and item recommendation [11]–[14]. However, some of these models break *causality*, in the sense that they either use the review of the item whose rate one is predicting, or use item reviews that have not been received by the time the item of interested was rated.

Finally, a model that combines RRN (a dynamical RS) with character-based autoregressive language models for reviews has recently been develop [15]. This work however, does not leverage the review content for rating prediction.

In contrast to all these works, we combine dynamical recommender systems with a dynamical language model that captures review content evolution, and use the review representations together with the user-item temporal representations in a causal fashion, to predict the rating of the next review.

III. Dynamic Review-based Recommenders (DRR)

The interests and preferences of users vary as they age, or change their social status or lifestyle. Exogenous factors like trends or seasons also affect user preferences. For example, users tend to look for different clothe types in winter than those they look for in summer. Users also tend to change their music tastes as they age. Such preference changes are naturally encoded in the collections of reviews and ratings given by these users over time. Our goal is to learn representations capturing them. We therefore develop a model that explicitly

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_10

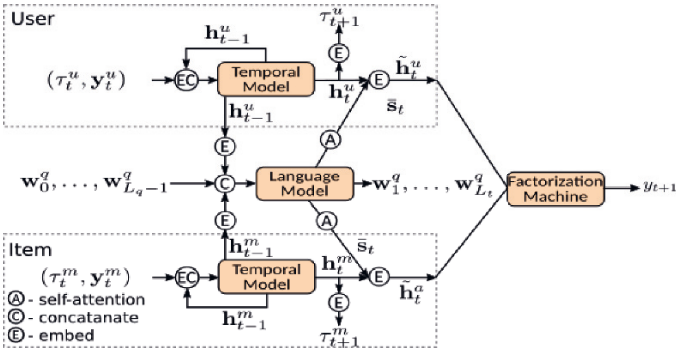


Fig. 1: Dynamic Review-based Recommender. The model consists of three interacting components: (i) a temporal model composed of two RNNs, one for users and the other for items, which we called *Dynamic Model of Review Sequences*; (ii) a neural language model which leverages the temporal representations of both user and items, and which we called *Dynamic Model of Review Content*; and (iii) a *Rating Model* which combines the user and item temporal representations with the review content representations to predict ratings. Note that when $q = t$ in the language model component, the Dynamic Review-based Recommender is causal. The model is non-causal when $q = t + 1$.

uses the text content and ratings of past reviews together with the history of when those reviews were written to better predict user interest in unknown items.

Consider a dataset $\mathcal{D}$ with a number of V items (as e.g. businesses or services, movies, products, etc.) and a number of U users. An element $e \in \mathcal{D}$ consists of a sequence of N_e reviews $\mathbf{r}_e = \{(\mathbf{x}^e_t, \tau^e_t, \delta^e_t, \mathbf{y}^e_t)\}^{N_e}_{t=1}$, where the t-th review is composed of its text $\mathbf{x}^e_t$, creation time τ^e_t, inter-review time $\delta^e_t \equiv \tau^e_t - \tau^e_{t-1}$ and rating vector $\mathbf{y}^e_t$.

Such review sequences $\mathbf{r}_e$ effectively define time series, and each of these can either be associate with a user u (in which case we set $e = u$), or an item v (in which case $e = v$).

Thus the rating vector for user u is such that $\mathbf{y}^u_t \in \mathbb{R}^V$, with $\mathbf{y}^u_{t,v} = p$ if user u rated item v with rating p. Conversely, the rating vector for item v is such that $\mathbf{y}^v_t \in \mathbb{R}^U$. Note that both of these vectors are large and sparse. To process them efficiently we perform dimensionality reduction via hashing, following [16].

Our main idea is to model the user and item review sequences separately, via two independent RNNs which output temporal representations encoding the nonlinear relations between timing and rating of past reviews. We then feed these temporal representations to neural models of text, thereby yielding instantaneous review content models, while simultaneously use them to predict when are new reviews going to arrive and what are their ratings. The model thus consists of tree interacting components: a temporal model composed of two RNNs, one for users and the other for items, which we called *Dynamic Model of Review Sequences*, a neural language model which leverages the temporal representations of both user and items, and which we called *Dynamic Model of Review Content*, and a *Rating Model* which combines the user and item temporal representations with the review content representations to predict ratings. In what follows we dwell into the details of these building blocks. Figure 1 summarizes the Dynamic Review-based Recommender (DRR) model.

A. *Dynamic Model of Review Sequences*

Given a sequence of reviews $\mathbf{r}_e$, we process each of its elements recursively via a RNN with hidden state $\mathbf{h}^e_t \in \mathbb{R}^H$. At each timestep t, we first compute the hidden representation

$$\mathbf{z}^e_t = \mathbf{W}^e_\tau \tau^e_t + \mathbf{W}^e_\delta \delta^e_t + \mathbf{W}^e_y \mathbf{y}^e_t + \mathbf{b}^e, \tag{1}$$

where $\mathbf{W}^e_\tau, \mathbf{W}^e_\delta, \mathbf{W}^e_y$ and $\mathbf{b}^e$ are learnable parameters and $\mathbf{z}^e_t \in \mathbb{R}^E$. We then update the RNN's hidden state thus

$$\mathbf{h}^e_t = f^{(e)}_\theta(\mathbf{z}^e_t, \mathbf{h}^e_{t-1}), \tag{2}$$

where $f^{(e)}_\theta$ is implemented by a LSTM network [17].

Note that the superindex e is used here to emphasize that we have two sets of functions namely, one for the user ($e = u$) and one for the item ($e = v$) reviews. The temporal representation $\mathbf{h}^e_t$ thus defined not only encodes the history of ratings, but also the time lag between past reviews, thereby yielding a continuous-time representation of the dynamics.

To enforce encoding quality, we first use $\mathbf{h}^e_t$ to predict the arrival time of new reviews via a simple Review Creation Model, which we shall now introduce. Later we will explicitly use $\mathbf{h}^u_t$ and $\mathbf{h}^v_t$ to predict ratings through a Rating Model.

1) Review Creation Model: The inter-review times δ^e_t can be modeled as following an exponential distribution whose rate parameter $\lambda^{(e)}_\theta(\mathbf{h}^e_t)$ is a function of the temporal representation $\mathbf{h}^e_t$ [18], [19]. In practice we approximate the function $\lambda^{(e)}_\theta : \mathbb{R}^H \to \mathbb{R}_{>0}$ with a multi-layer perceptron. The log-likelihood of the Review Creation Model is then

$$\begin{aligned}\log p(\delta^e) &= \sum_{t=1}^{N_e} \log p_\theta(\delta^e_{t+1} | \mathbf{h}^e_t) \\ &= \sum_{t=1}^{N_e} \left(\log \lambda^{(e)}_\theta (\mathbf{h}^e_t) - \lambda^{(e)}_\theta (\mathbf{h}^e_t)\, \delta^e_{t+1} \right).\end{aligned} \tag{3}$$

Note that predicting the arrival times of new reviews can be done by either sampling the exponential distribution, or using the mean of the distribution directly. In our experiments we use the mean of the distribution.

B. Dynamic Model of Review Content

Consider the t-th review in the sequence $\mathbf{r}_e$, whose text content is given by $\mathbf{x}_t^e = (\mathbf{w}_0^{e,t}, \mathbf{w}_1^{e,t}, \dots, \mathbf{w}_{L_t^e}^{e,t})$, where $\mathbf{w}_j^{e,t}$ and L_t^e label the j-th word and the number of words in that review, respectively. To capture how the review content changes within $\mathbf{r}_e$, we define the probability of observing the word sequence $\mathbf{x}_t^e$ at the t-th review as the conditional probability $p(\mathbf{x}_t^e|\mathbf{h}_{t-1})$. Here we define the global temporal representation $\mathbf{h}_t$ encoding the nonlinear relations between timing and ratings of past reviews as $\mathbf{h}_t \equiv \text{concat}([\mathbf{h}_t^u, \mathbf{h}_t^v])$, with $\mathbf{h}_t^{u,v}$ defined in Eq. 2.

Note that when processing the dataset $\mathcal{D}$, the modeling of review content does not need to differentiate between user and item. We therefore drop the superindex e in what follows.

Below we present two models for $p(\mathbf{x}_t|\mathbf{h}_{t-1})$, one based on a Bag-of-Words (BoW) representation, and another on an autoregressive language model. Both models will be trained by maximising $\log p(\mathbf{x}_t|\mathbf{h}_{t-1})$. These language models will ultimately allow us to define a vector representation $\bar{\mathbf{s}}_t$, summarizing the content of the t-th review, which we will later use as input to our Rating Model.

1) Bag-of-Words Neural Review Model: We assume the words in $\mathbf{x}_t$ are generated independently, conditioned on $\mathbf{h}_{t-1}$, that is $p(\mathbf{x}_t|\mathbf{h}_{t-1}) = \prod_j^{L_t} p_\theta(\mathbf{w}_j^t|\mathbf{h}_{t-1})$, where we follow [20] and write the probability over words as

$$\begin{aligned} p_\theta(\mathbf{w}_j^t|\mathbf{h}_{t-1}) &= \frac{\exp\{-a(\mathbf{w}_j^t, \mathbf{h}_{t-1})\}}{\sum_{k=1}^V \exp\{-a(\mathbf{w}_k^t, \mathbf{h}_{t-1})\}}, \\ a(\mathbf{w}_j^t, \mathbf{h}_{t-1}) &= -\mathbf{h}_{t-1}^\top \mathbf{R}\, \mathbf{w}_j^t - \mathbf{b}\, \mathbf{w}_j^t, \end{aligned} \tag{4}$$

with $\mathbf{R} \in \mathbb{R}^{2H \times V}$ and $\mathbf{b} \in \mathbb{R}^V$ trainable parameters, $\mathbf{h}_t = \text{concat}([\mathbf{h}_t^u, \mathbf{h}_t^v])$ and $\mathbf{w}_j^t$ the one-hot representation of the j-th word in $\mathbf{x}_t$.

We define the summary representation for $\mathbf{x}_t$ as the Bag-of-Words (BoW) representation $\bar{\mathbf{s}}_t \in \mathbb{R}^V$, where V is the vocabulary size [21].

2) Autoregressive Review Model: In contrast to the BoW model above, autoregressive language models approximate the probability over the word sequence $\mathbf{x}_t$ as [22]

$$p(\mathbf{x}_t|\mathbf{h}_{t-1}) = \prod_{j=1}^{L_t} p_\theta(\mathbf{w}_j^t|\mathbf{w}_{<j}^t, \mathbf{h}_{t-1}), \tag{5}$$

where $\mathbf{w}_{<j}^t$ labels all words previous to $\mathbf{w}_j^t$.

The conditional probability above depends on both $\mathbf{w}_{<j}^t$ and the global temporal representation $\mathbf{h}_t$. To model it we take an approach akin to that of the variational autoencoders of text [23]. That is, we first concatenate $\mathbf{h}_t$ with all word embeddings in $\mathbf{x}_t$, i.e. we define $\tilde{\mathbf{w}}_j = \text{concat}[\mathbf{w}_j, \mathbf{h}_{t-1}]$, and then process the new vector sequence with a RNN with hidden state $\mathbf{s}_k^t \in \mathbb{R}^S$, whose update equation reads $\mathbf{s}_j^t = g_\theta(\tilde{\mathbf{w}}_j, \mathbf{s}_{j-1}^t)$. Here g_θ is implemented by a LSTM network, with equations similar to those below Eq. (2).

The distribution p_θ is then defined as a categorical distribution over a vocabulary of size V, whose class probabilities are given by $\boldsymbol{\pi}_j^t = \text{softmax}(\mathbf{W}\, \mathbf{s}_j^t)$, where $\mathbf{W} \in \mathbb{R}^{V \times S}$ is a learnable matrix.

We now define the summary representation for $\mathbf{x}_t$ as a weighted sum over word representations $\bar{\mathbf{s}}_t = \sum_j^{L_t} \alpha_j^t \mathbf{s}_j^t$ where the j-th weight α_j^t is calculated with the *gated attention mechanism* proposed in [24]

$$\begin{aligned} \alpha_j^t &= \text{softmax}(\mathbf{k}_j^\top \mathbf{q}), \\ \mathbf{k}_j &= \tanh(\mathbf{M}_1\, \mathbf{s}_j^t + \mathbf{b}_1) \odot \sigma(\mathbf{M}_2\, \mathbf{s}_j^t + \mathbf{b}_2), \end{aligned} \tag{6}$$

where $\mathbf{M}_1, \mathbf{M}_2 \in \mathbb{R}^{A \times S}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^A$ are learnable parameters, $\mathbf{q} \in \mathbb{R}^A$ can be interpreted as a learnable global query, $\odot$ denotes element-wise multiplication and $\sigma(\cdot)$ denotes the sigmoid function. This type of attention is introduced to solve the problem of the limited expressiveness of the $\tanh(\cdot)$ to capture complex relations, due to the fact of approximate linearity in the region $[-1, 1]$.

As we shall see below, this attentive summary representation allows us to track the most relevant words affecting the rating of a given item as time evolves.

C. Combining temporal and summary representations

Given the temporal representations for user and item reviews (i.e. $\mathbf{h}_t^u$, $\mathbf{h}_t^v$), and the summary representation for review content $\bar{\mathbf{s}}_t$, we want to predict the rating $\hat{y}_t^{uv} \in \mathbb{R}$ that user u gives to item v. There is, however, still the question of how to combine $\mathbf{h}_t^u$ and $\mathbf{h}_t^v$ with $\bar{\mathbf{s}}_t$. After exploring different possibilities we found two optimal solutions namely,

1) DRR-BoW: For the Bow Neural Review Model we augment Eq. (1) and define $\tilde{\mathbf{z}}_t^e = \mathbf{z}_t^e + \mathbf{W}_s^e\, \bar{\mathbf{s}}_t$, where $\mathbf{W}_s^e \in \mathbb{R}^{H \times S}$ is an additional learnable weight, to get $\tilde{\mathbf{h}}_t^e = f_\theta^{(e)}(\tilde{\mathbf{z}}_t^e, \tilde{\mathbf{h}}_{t-1}^e)$, where $f_\theta^{(e)}$ remains the same as in Eq. (2). The new representation $\tilde{\mathbf{h}}_t^e$ now encodes the nonlinear interaction between timing, rating *and text* of past reviews.

2) DRR-LM: For the Autoregressive Review Model we instead define

$$\tilde{\mathbf{h}}_t^e = \mathbf{W}^{(e)}\text{concat}([\mathbf{h}_t^e, \bar{\mathbf{s}}_t]) + \mathbf{b}^{(e)}, \tag{7}$$

with $\mathbf{W}^{(e)} \in \mathbb{R}^{H \times (H+S)}, \mathbf{b}^{(e)} \in \mathbb{R}^H$ learnable. The resulting representation $\tilde{\mathbf{h}}_t^e$ also encodes the interaction between timing, rating and text, albeit through a different route.

D. Rating Model

We have now all ingredient to predict the rating $\hat{y}_t^{uv} \in \mathbb{R}$ that user u gives to item v. We compute $\hat{y}_t^{uv}$ with a factorization machine (FM) [25], here defined as

$$\hat{y}_{t+1}^{uv}(\mathbf{h}) = w_0 + \sum_{i=1}^{2H} w_i h_i + \sum_{i=1}^{2H} \sum_{j=i+1}^{2H} \langle \mathbf{v}_i, \mathbf{v}_j \rangle h_i h_j, \tag{8}$$

where $\mathbf{h} \in \mathbb{R}^{2H} \equiv \text{concat}([\tilde{\mathbf{h}}_t^u, \tilde{\mathbf{h}}_t^v])$, $w_0 \in \mathbb{R}$, $\mathbf{w} \in \mathbb{R}^{2H}$ and $\mathbf{V} \in \mathbb{R}^{2H \times K}$ are learnable parameters, K is set to 10 and $\langle \cdot, \cdot \rangle$ denotes dot product.

We choose the loss function of the Rating Model to be the mean square error function between $\hat{y}_t^{uv}$ and our prediction $\hat{y}_t^{uv}(\mathbf{h})$.

E. DRR Loss function

The complete loss function of the DRR model has therefore three components: the loss of the Rating Model, the loss of the Dynamic Model of Review Sequences, which is the negative log-likelihood of an exponential, and the loss of the Dynamical Model of Review Content, which is the negative log-likelihood of our word sequence model. Explicitly we write

$$\begin{aligned}\mathcal{L} =& \frac{1}{|\mathcal{D}|}\sum_{u,v\in\mathcal{D}}\sum_{t}(y_t^{uv} - \hat{y}_{t=1}^{uv})^2 \\ & -\lambda_1 \sum_{e\in\mathcal{D}}\sum_{t=1}^{N_e}\log p_\theta(\delta_{t+1}^e|\mathbf{h}_t^e) - \lambda_2\sum_t \log p_\theta(\mathbf{x}_t|\mathbf{h}_{t-1}),\end{aligned} \tag{9}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}^+$ are hyperparameters.

IV. Causality

By construction, both DRR-BoW and DRR-LM models above preserve causality — the models do not use *any* information from the future to predict ratings. As mentioned in the introduction, however, most recommender system models that leverage review content use the review $\mathbf{x}_{t+1}^v$, written by user u, to predict the rating y_{t+1}^{uv} given by this same user to the item v. In order to fairly compare our methodology with such models, we use the degrees of freedom available within the definition of the DRR-LM model and redefine

$$\tilde{\mathbf{h}}_t^e = \mathbf{W}^{(e)}\text{concat}([\mathbf{h}_t^e, \bar{\mathbf{s}}_{t+1}]) + \mathbf{b}^{(e)}. \tag{10}$$

This new representations encodes $\bar{\mathbf{s}}_{t+1}$, the summary representation of the review whose rating it predicts, and breaks causality. Below we refer to the model using the causal representation Eq. (7) as DRR-LM-C, whereas we denote the model using the non-causal expression Eq. (10) as DRR-LM-NC.

Naturally, the causal model is to be preferred as we normally do not have review content about the item whose rating we want to predict. Nevertheless, we shall see that the non-causal model lends itself when one is interested in tracking the words which most affect the rating of a given item as time evolves.

V. Experiments and Results

Data set To test our model we choose the Amazon dataset [26]. We pick four 5-core subcategory datasets namely, *Automotive (A)*, *Digital Music (DM)*, *Tools and Home (TH)* and *Pet Supplies (PS)*. The review creation time is defined as the difference in days between the original timestamp and the timestamp of the first review in the dataset. Next we group reviews by day, since the granularity of the timestamps is day based. All users or items with less than 5 days (i.e. time series with less than 5 points) are removed from the dataset. The autoregressive language models use the review raw text, changed into lower case. Preprocessing scripts can be found at [1]. Statistics of the preprocessed data is summarized in Table I.

Training Our model predicts ratings through the user and item dynamic representations, which come from two independent RNNs. Simply applying backpropagation through both sequences is computationally forbidden. In order to overcome this problem, we train the user and item RNNs alternately. We first freeze the parameters of e.g. the items' RNN, and only update those of the users' RNN, while back-propagating the gradients of all ratings for a user batch. The items' dynamic representations are taken to be fixed. We then repeat these operations but now with the user parameters and user representations frozen.

Model Configuration We split each dataset along the time dimension into three parts: training set (80%), validation set (10%) and test set (10%). We use grid search on the validation set for hyperparameter tuning. We set the hidden dimension H of the temporal representation $\mathbf{h}_t^e$ to 32, and the embedding dimension E of $\mathbf{z}_t^e$ to 100. Regarding the review content models, we set the vocabulary size V to 2000 for DRR-BoW and to 5000 for DRR-LM. In the latter case we also use GloVe word embeddings [27] (these corresponds to the $\mathbf{w}_j^t$ in Eq. (5)) with dimension 300. For DRR-LM we also set the attention dimension A to 64 and the embedding dimension H' of the (concatenation of the) temporal and summary representations to 64. We use Adam [28] with learning rate 0.0002 and $\beta_1 = 0.9$ and limit the review length to 150 tokens. All methods are implemented using PyTorch v1.3[1]. Source code for all models can be found at [1].

Results Given an user and item of interest, the DRR model predicts the arrival time, rating and the probability over the word sequence of the next review, and we optimize the model to give the best performance on the rating prediction task.

Our methodology incorporates modeling the dynamic aspects of user-item interaction with neural models of review content. To test the importance of each of these components for the problem of rating prediction, we test our models against (i) the Probabilistic Matrix Factorization (PMF) [29], which is a static recommender system which does not model review content; (ii) the RRN [10], a causal model which learns dynamic user/item representations (albeit non-continous), but does not model review content; and (iii) three static models which do leverage review content, namely DeepCoNN [14], D-ATT [13] and AHN [12]. These last three models are non-casual since they either use the review of the item whose rate they predict, or use item reviews that have not been received by the time the item of interest was rated. Table II shows results for all models on the chosen datasets. We use boldface to highlight best results in both causal and non-causal cases.

Let us start by focusing on the causal models. First we note that both DRR-BoW and DRR-LM-C outperform the RRN model, which confirms the known fact that review content helps in rating prediction tasks. We remark however that in this case the models in questions are dynamic, and it is the content of *past reviews* what is successfully being used. Interestingly, DRR-BoW beats DRR-LM-C which may hint at the fact that

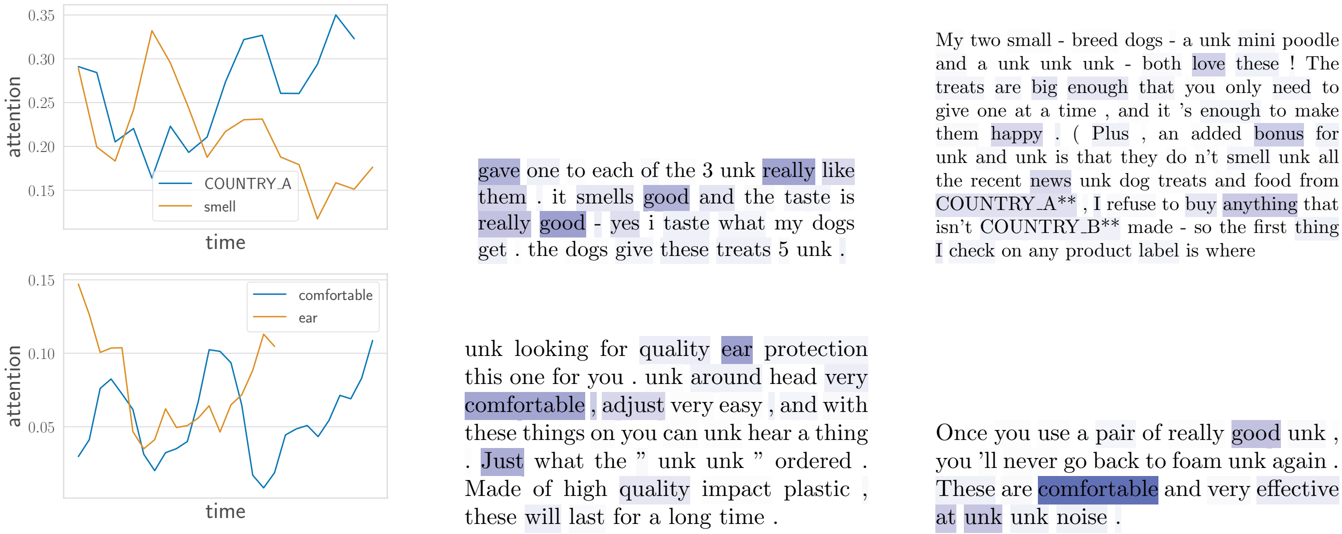
[1] https://pytorch.org/

TABLE I: Datasets statistics. The mean and the standard deviation of the number of reviews, sentences and words per review with respect to the user and item.

	Automotive		Digital Music		Tools and Home		Pet Supplies	
	user/item		user/item		user/item		user/item	
	mean	std	mean	std	mean	std	mean	std
reviews	6.1/9.3	1.7/5.5	7.8/9.0	6.2/6.5	7.8/10.6	5.3/9.0	7.44/13.7	4.4/14.2
sentences	8.7/9.7	6.3/7.5	6.3/4.9	11.9/10.2	8.4/7.5	8.1/7.8	7.8/6.5	7.7/6.9
words	89.8/101.6	68.3/82.5	52.5/38.2	106.4/94.6	80.3/70.5	85.6/83.3	68.9/55.9	71.1/66.2

TABLE II: Mean-square error on the rating prediction (* results taken from [12]).

	static	non-causal models				causal-models		
Datasets	PMF*	DeepCoNN*	D-ATT*	AHN*	DRR-LM-NC	RRN	DRR-BoW	DRR-LM-C
A	0.9187	0.7809	0.7654	**0.7314**	0.7791	1.0927	**0.7838**	0.8171
DM	0.8788	0.8754	0.8506	0.8172	**0.7250**	0.7961	**0.7723**	0.7801
TH	1.1182	0.9856	0.9850	0.9671	**0.9264**	1.0896	**1.0406**	1.0656
PS	1.4340	1.2598	1.2730	1.2515	**1.0500**	1.1970	**1.1734**	1.1918



gave one to each of the 3 unk really like them . it smells good and the taste is really good - yes i taste what my dogs get . the dogs give these treats 5 unk .

My two small - breed dogs - a unk mini poodle and a unk unk unk - both love these ! The treats are big enough that you only need to give one at a time , and it 's enough to make them happy . (Plus , an added bonus for unk and unk is that they do n't smell unk all the recent news unk dog treats and food from COUNTRY_A** , I refuse to buy anything that isn't COUNTRY_B** made - so the first thing I check on any product label is where

unk looking for quality ear protection this one for you . unk around head very comfortable , adjust very easy , and with these things on you can unk hear a thing . Just what the " unk unk " ordered . Made of high quality impact plastic , these will last for a long time .

Once you use a pair of really good unk , you 'll never go back to foam unk again . These are comfortable and very effective at unk unk noise .

Fig. 2: *Upper Left:* Dynamic attention on the words 'COUNTRY_A' and 'smell' for an item in 'Pet Supplies' dataset. *Upper Middle:* Review sample from the beginning of the time series. *Upper Right:* Review sample from the end of the time series. **The real names of the countries are replaced with masks 'COUNTRY_A' and 'COUNTRY_B' for fairness. *Lower Left:* Dynamic attention on the words 'comfortable' and 'ear' for an item in the 'Tools and Home' dataset. *Lower Middle:* Review sample from the beginning of the time series. *Lower Right:* Review sample from the end of the time series. The darker the highlight color for a word, the higher its attention value.

it is enough to know that certain key words are present in the review, as opposite to e.g. word order, to better predict the rating. Regarding the non-causal models, DRR-LM-NC outperforms all other models in almost all datasets, which shows that one indeed needs to not only account for review content, but also for its dynamic character. Remarkably, both causal models DRR-BoW and DRR-LM-C perform better than all their non-causal competitors in two of the datasets (see the Digital Music and Pet Suplies rows in the table), and comparable to them in the others.

We can conclude that our models successfully learn both temporal user/item representations and review content representation which *together* are useful for rating prediction.

Let us now consider the dynamic attention mechanism of the DRR-LM-NC, which allows us to e.g. follow in time the weights α_j^t (defined in Eq. 6) of the words in the reviews for the item whose rate we aim at predicting. The higher the weight of a word, the stronger its relevance to the rating prediction. Figure 2 *Upper Left* shows the attention weights on the words 'COUNTRY_A' and 'smell' as time evolves for a given product in the 'Pet Supplies' dataset. One can see that although at the start of the time series the word 'smell' was important for determining the rating, its relevance decreases as the weight on the word 'COUNTRY_A' increases. After a closer look at the reviews we learn that at the start of the time series most reviews were related to the smell of the product (e.g. whether the dogs were liking the product's smell). Later on, however, the manufacturing company moved the product production to COUNTRY_A, and this event was successfully captured by our attention model. Figure 2 *Upper Middle* shows an example review for the item in question, from the start of the time series. Words with darker highlights mean here words with higher attention weight. One can see that the word 'smell' is highlighted as important. In contrast, Figure 2 *Upper Right* displays a review sampled from the end of the time series, in which one sees the word 'COUNTRY_A' has more relevance than the word 'smell'. Similarly, the *Lower* row of Figure 2 shows the attention weights on the words 'comfortable' and 'ear' as time evolves for a given product in the 'Tools and Home' dataset.

VI. Conclusion and Feature Work

In this work we proposed a recommender system model which accounts for the dynamic aspects of user preferences, as reflected in their history of reviews and ratings. We explicitly learn continuous-time representations for both users and items, and use these to define dynamic language models for review content. The latter provided us with review content representations which, when combined with the temporal user/item representations, proved to be useful in predicting the hidden interest of users in unknown items. Indeed, our results outperformed several state-of-the-art recommender system models in rating prediction tasks, in different datasets.

We also introduced a new dynamic attention mechanism which allowed us to track the most relevant words for a given rating of an item of interest at a given instant of time.

Future directions of work include developing attention mechanisms between reviews with different timestamps, and learning more generic dynamic representations able to characterize hidden dynamics global to all users.

Acknowledgment

The authors of this work were supported by the Competence Center for Machine Learning Rhine Ruhr (ML2R) which is funded by the Federal Ministry of Education and Research of Germany (grant no. 01—S18038A). We gratefully acknowledge this support.

References

[1] "Source code," https://figshare.com/s/2e19d38501d275944487.

[2] Y. Koren, "Collaborative filtering with temporal dynamics," *Commun. ACM*, vol. 53, no. 4, p. 89–97, Apr. 2010. [Online]. Available: https://doi.org/10.1145/1721654.1721677

[3] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[4] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted boltzmann machines for collaborative filtering," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 791–798.

[5] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111–112.

[6] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," *arXiv preprint arXiv:1511.06939*, 2015.

[7] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 130–137.

[8] Y. K. Tan, X. Xu, and Y. Liu, "Improved recurrent neural networks for session-based recommendations," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, 2016, pp. 17–22.

[9] B. Twardowski, "Modelling contextual information in session-aware recommender systems with neural networks," in *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016, pp. 273–276.

[10] C.-Y. Wu, A. Ahmed, A. Beutel, A. J. Smola, and H. Jing, "Recurrent recommender networks," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 495–503.

[11] R. Catherine and W. Cohen, "Transnets: Learning to transform for recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*. ACM, 2017, pp. 288–296.

[12] X. Dong, J. Ni, W. Cheng, Z. Chen, B. Zong, D. Song, Y. Liu, H. Chen, and G. de Melo, "Asymmetrical hierarchical networks with attentive interactions for interpretable review-based recommendation," *ArXiv*, vol. abs/2001.04346, 2019.

[13] S. Seo, J. Huang, H. Yang, and Y. Liu, "Interpretable convolutional neural networks with dual local and global attention for review rating prediction," in *Proceedings of the eleventh ACM conference on recommender systems*, 2017, pp. 297–305.

[14] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 425–434.

[15] C.-Y. Wu, A. Ahmed, A. Beutel, and A. J. Smola, "Joint training of ratings and reviews with recurrent recommender networks," 2016.

[16] K. Weinberger, A. Dasgupta, J. Langford, A. Smola, and J. Attenberg, "Feature hashing for large scale multitask learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 1113–1120.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[18] K. Cvejoski, R. J. Sanchez, B. Georgiev, J. Schuecker, C. Bauckhage, and C. Ojeda, "Recurrent point processes for dynamic review models," in *Workshop on Interactive and Conversational Recommendation Systems at AAAI*, 2020.

[19] K. Cvejoski, R. J. Sánchez, B. Georgiev, C. Bauckhage, and C. Ojeda, "Recurrent point review models," in *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–8.

[20] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *International conference on machine learning*, 2016, pp. 1727–1736.

[21] G. E. Hinton and R. R. Salakhutdinov, "Replicated softmax: an undirected topic model," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1607–1614. [Online]. Available: http://papers.nips.cc/paper/3856-replicated-softmax-an-undirected-topic-model.pdf

[22] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, "Recurrent neural network based language model." in *INTERSPEECH*, T. Kobayashi, K. Hirose, and S. Nakamura, Eds. ISCA, 2010, pp. 1045–1048.

[23] S. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 10–21.

[24] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based deep multiple instance learning," *arXiv preprint arXiv:1802.04712*, 2018.

[25] S. Rendle, "Factorization machines," in *2010 IEEE International Conference on Data Mining*. IEEE, 2010, pp. 995–1000.

[26] R. He and J. McAuley, "Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering," in *proceedings of the 25th international conference on world wide web*, 2016, pp. 507–517.

[27] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.

INDUSTRY TRACK

Abstracts

Text Based Category Code Machine Learning Classification for Smart Procurement

O. Ugur, A. A. Arısoy, M. Gulcakir and M. C. Ganiz

In a construction project, the procurement operation is crucial in terms of project cost. Especially for large scale projects the size of this operation can become very large as well, in terms of the number and variety of purchased items. ENKA Systems, which is a subsidiary of the ENKA, the largest construction company in Turkey, has developed a software system for procurement processes which includes a coding systems to standardize the procurements that is especially beneficial for large scale projects. As the system is kept being used, the hierarchical tree structured coding reached a fairly complicated system due to its extensive scope. This causes the procurement officials to oversee a large part of the operations manually to make sure that the properties of the items are correctly inputted so that they are traceable, for future purchases as well. In this study a machine learning model for classifying the category code of the purchased item is developed. The quantity, unit of measurement, description and common cost code, which is an important input according to the information provided by the ENKA's procurement and supply chain management department, are used as input for the developed models. The models are trained with raw text data, text data with pseudo word inputs such as "__UNIT__" and "__QUANTITY__", and using the text information of the item description combined with one-hot encoded format of unit and common cost code parameters. The performance of these models are compared and results are verified using 10-fold cross validation method and inputs provided by the procurement officials.

Scaling Artificial Intelligence in Industrial Applications

F. Niszl, S. Meixner, A. Suendermann, J. Kemnitz and D. Schall

Artificial Intelligence (AI) is increasingly explored in various domains and industries. Many companies experiment with AI, but too often those experiments are one-off analyses based on outdated data and the resulting models never make it into production. Therefore, we suggested to structure the AI model into a predefined template and enable a self-serviced app a domain expert can operate via a dashboard. The operator can link recorded data to the model, trigger the training, check quality metrics, and deploy the model in "one-click" on the target platform as for example an edge device. The edge device links sensor data with the model input and model output as feedback back into the industrial process. Model training, deployment and lifecycle management can be carried out in a scalable manner by a non-expert. A large number of models can be managed in parallel, and data can be linked to the respective sensor or machine. This approach is generic to a large set of typical sensor data. In this work we show the application in two different use-cases i) visual quality inspection in an assembly line and ii) anomaly detection in time series data.

Data ecosystems - An incubator for data innovation

L. Höllbacher

We are witnessing a new industrial revolution driven by data, computation and automation. Data has thus become a driver of economic success, but it‘s full potential only unfolds through cross-company collaboration that simultaneously ensures control over proprietary data across corporate boundaries. This requires a powerful and competitive as well as secure and trustworthy infrastructure that is compatible with the European GAIA-X initiatives. This is exactly the solution nexyo offers: an operational IT system for decentralised data networks that enables scalable data governance and cross-company data exchange.

Hybrid Method for Targeted Conversations in Online Classified Marketplace

Y. Rahimi, A. Kamandi, A. Hoseini and H. Haddad

Online/offline chat is a convenient approach in the electronic markets of second-hand products in which potential customers would like to have more information about the products to fill the information gap between buyers and sellers. In this article, we introduce a method for the question / answer system that we have developed for the top-ranked electronic market in Iran called Divar which is in top 20 classified sites in the world by semi supervised and distributed system. When it comes to secondhand products, Incomplete product information in a purchase will result in loss to the buyer. One way to balance buyer and seller information of a product, is to help the buyer ask more informative questions when purchasing. Also, the short time to start and achieve the desired result of the conversation was one of our main goals, which was achieved according to A/B tests results. Profit of creating such systems is to help users gather knowledge about the product easier and faster. We collected a data set of around 10 million messages in Persian colloquial language and for each category of product we gathered 1000K messages, of which only 2K were Tagged and semi-supervised methods were used. In order to publish the proposed model to production, it is required to be fast enough to process each conversation in micro second on CPU processors. In order to reach that speed, in many sub tasks faster and simplistic models are preferred over deep neural models and for handling this we proposed distributed semi supervised method which requires only a small amount of labeled data and by utilizing text categorization methods to reach the information available for each messages. Currently our model used in Divar production on CPU processors and 15% of buyers and seller’s messages in conversations is directly chosen from model output and more than 27% of buyers have used this model suggestions in at least one daily conversation.

Provided Papers - Non Reviewed

Beyond Desktop Computation: Challenges in Scaling a GPU Infrastructure

Martin Uray*, Eduard Hirsch*, Gerold Katzinger† and Michael Gadermayr*
*Salzburg University of Applied Sciences, 5412 Puch, Austria
†NTS Netzwerk Telekom Service AG 5020 Salzburg, Austria
{martin.uray, eduard.hirsch, michael.gadermayr}@fh-salzburg.ac.at,
gerold.katzinger@nts.eu

***Abstract*—Enterprises and labs performing computationally expensive data science applications sooner or later face the problem of scale but unconnected infrastructure. For this up-scaling process, an IT service provider can be hired or in-house personnel can attempt to implement a software stack. The first option can be quite expensive if it is just about connecting several machines. For the latter option often experience is missing with the data science staff in order to navigate through the software jungle. In this technical report, we illustrate the decision process towards an on-premises infrastructure, our implemented system architecture, and the transformation of the software stack towards a scaleable Graphics Processing Unit (GPU) cluster system.**

***Index Terms*—Shared Computing, GPU, Infrastructure, On-Premises, Cloud**

I. Introduction

In the course of computing history, sufficient computing power often exhibited a basis for the application of novel ideas. But it often also has been the basis for research ideas that were not superior because of the idea, but simply because of sufficient computing power available. Here, in some kind of sense, the increasing computational power drove the development of research ideas [1]. The steadily increasing amount of computational power was long driven by Moore's Law, whereas the producing industry is not driven by this observation anymore [2]. Also the field of Artificial Intelligence (AI) gained popularity due to the amount of increased computational power, and evolved in new fields, like Deep Learning (DL) [3], [4].

Due to the promising results by applications of AI methods, a lot of research is currently performed in the field of AI based methods. Working with AI, especially DL, the question of the computational resources arises sooner or later. We observed, that in a lot of companies, Data Scientists are often the only ones with expertise in Computer Science. Besides their main function, they have to take care about computing infrastructure - even if it is not their field of expertise. This causes, that a huge amount of time is spent on research for applicable tools, systems, and environments to develop on.

Similar to other companies, start-ups, and universities, a simple GPU on-premises infrastructure is maintained within our institute, consisting of only a single server. This computing infrastructure was initially implemented as simple as possible, due to the lack of expertise and time. By using fair-share, one server with eight GPUs was set up. This machine specification can be found in Section IV, line C_1. On this machine, for eligible people access was granted for research and training. When submitting a computation job, one user had to choose a not used GPU and start the job by only using this one dedicated resource. For multi-GPU jobs, several resources were allocated and blocked. With this setup it was likely to happen, that one user did not restrict a jobs resources according to the policy and blocked all computational resources. Additionally, inferences with other jobs are possible, leading to failures in all involved jobs. With a growing number of users on such a shared hardware, the demand for manageable permissions and restrictions increased.

Up-scaling and extending such an on-premises infrastructure, while preserving easy usage can be quite complex, without the help of experts in the field of IT infrastructure and High-Performance Computing. Also, the decision, on whether to move to the cloud shall be well defined, since this must be argued towards various stakeholders.

In the progress of planning the extension of our existing computational resources, several questions came up while being aligned with the set requirements:

Q1 Is a transition into the cloud financially beneficial or should the existing on-premises infrastructure be extended?
Q2 Are there preexisting solutions available?
Q3 What hardware components are needed and how to design the architecture?

Martin Uray and Eduard Hirsch were supported by the European Interreg Österreich-Bayern project AB215 DataKMU and Michael Gadermayr was supported by the County of Salzburg under grant number FHS-2019-10-KIAMed.

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_11

Q4 What software components need to be part within the used software stack?

In this work, we are outlining the architecture of our established GPU computing infrastructure, as it scaled from a single server to a multi-instance computing cluster. This is based on the decision towards an on-premises infrastructure and the alignment with defined requirements. This report shall not give the impression of being a best practice, however, it is intended to show the considerations that are necessary when transforming infrastructure to a multi-instance cluster setup.

This report is structured as follows: In Section II we give a short overview of our considerations to decide on a Cloud or on-premises computing infrastructure within our institution. Section III outlines the requirements on the infrastructure, that led to the architecture of the infrastructure Section IV and the setup of the cluster Section V. In Section VII we discuss about advantages and disadvantages of our implementation with an outlook and further considerations.

II. Cloud vs. On-premises Computing

Deciding on weather to run ones projects in the cloud or as an on-premise infrastructure involves many different aspects which are examined in the following. As one question this report shall answer is weather a transition for the computational tasks into the cloud is beneficiary, the emphasis in this section is put on factors which could lead or force ones intention to one or the other solution.

Like described in [5], there are models for grasping the scope of IT activities which shall not be our main focus here. Nevertheless, finding an appropriate solution for the own institution or company is key to a cost-effective solution. When further referring to [5] not just IT based Operations and Infrastructure (O&I) need to be considered but also ones that influence those. Thus, each of these may change when adopting from one model to another. Due to these manifold influencing factors in those areas and activities, a decision is highly based on the processes an institution applies.

Some of the important key-factors are: Cost, Scalability / Upgradeability, Network Connectivity, Maintainability, Security, General Data Protection Regulation, Disaster Assistance, and Data Backup and Recovery. These factors are essential to consider when planning to deploy services to cloud providers. Nevertheless, *cost* is often the one considered first.

A. Costs

When comparing Cloud to On-Premise expenses, companies usually start looking at hardware bought for local usage and on-demand (virtual) hardware. But it is necessary to get a complete picture of the total costs which cannot be reduced to a plain procurement process.

As indicated in [5], O&I needs to be accounted for which translate to Capital Expenditures (CapEx) and Operation Expenditures (OpEx) when referring to economical terms. Additionally, it is noted, that especially for maintenance, it is necessary to spend a regular, potentially high amount for an on-premise solution compared to a similar cloud resource. This is based on the companies needs: electricity costs, administrative staff, licenses and trainings.

TABLE I
Rough estimate in EUR when On-Premise solutions pay off.[2]

Month	**1**	**2**	**3**	**11**	**16**
Procurement	24500				
Electricity	250	250	250	250	250
Manpower	1200	1200	30	30	30
Cost p. m. On-Premise	25950	1450	280	280	280
Overall On-Premise	25950	27400	27680	**29920**	**31320**
Cost p. m. Google	2057	2057	2057	2057	2057
Overall Google	2057	4114	6171	22627	**32912**
Cost p. m. Azure	2947	2947	2947	2947	2947
Overall Azure	2947	5895	8842	**32420**	47156

Cloud resources, therefore, are incorporating those additional costs, offering services at a fixed price per resource and consumed time. However, it is still possible, to create solid cost efficient structures for machine learning when reducing OpEx and keeping CapEx at an acceptable level. Especially when dealing with *Green AI*. That may be the usage of renewable energy sources or facilitating outdated refurbished machines/hardware, in order to establish a GPU cluster.

For the decision on the extension of the institutes infrastructure experiments, regarding the usage of the existing infrastructure, where conducted. Over the period of four months (April - July 2021) the usage was monitored and logged. The usage over this period is used as a baseline[1] for the calculation process. Overall 7366 hours of GPU usage were monitored, what translates to a 32% grade of operation.

In Table I an estimation of the costs over a period of 16 months is shown. The procurement costs are based on the existing on-premise infrastructure (C_1). The costs of electricity are estimated based on the assumption, that 60% of the maximum power consumption is used in idle mode and the rest added based on the actual usage.[3] The assumption on manpower is based on the fact, that the initial setup was done during the first two months and further maintenance was barely needed. With this on-premise solution, the initial costs were high, but the overall monthly costs were kept low.[4]

For the calculation of the monthly costs, the mean monthly usage is considered (1843h/month). For both, MS Azure and Google Cloud, a setup was chosen with a commitment to min. three years.[5]

Comparing all three variants (on-premises, Google Cloud and Microsoft Azure) with the observed rate of operation, on the "costs per month" the Azure configuration has a break-even at month 11 and the one from Google Cloud from month

[1] Pessimistic baseline, since during the second half of the year (Aug - Dec) a much higher usage is observed.

[2] Rounded to full numbers. Break-even points indicated in bold.

[3] Not covered in calculations: 5% of the electricity consumed in-house are produced with solar panels.

[4] The hourly rate for manpower has been set to 30€/h. The server is running at 2600 Watt and the electricity prices set at 0,28€/kWh.

[5] Prices (in EUR) as of 05.08.2021. Azure configuration: *NC24s v3* instances with 24vCPU, 448 GiB Ram, 4x Tesla V100. Google Cloud: *AI platform* with *BASIC_GPU* training tier. All platforms hosted in Europe.

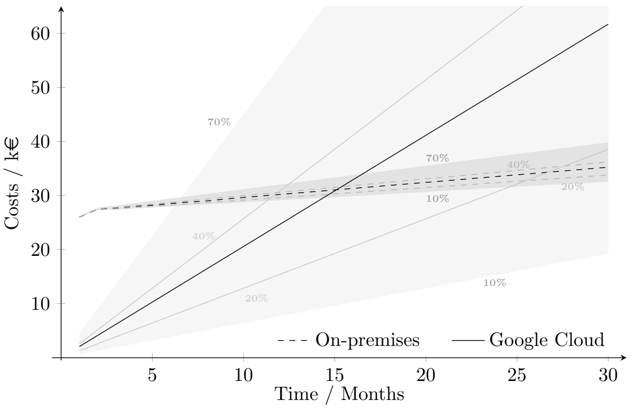


Fig. 1. Cost tendency graph, illustrating the costs per usage for the used on-premises (dashed line) and a fictional cloud solution (solid line) for the calculated usage of the cluster. The shadowed areas denote varying usage in the range of $10 - 70\%$. For each, on-premises and Google Cloud, a rough estimation for the usage of 20% and 40% is indicated.

16. Although, these systems are not completely equal in terms of hardware, the table provides a rough direction when an on-premise solution gets profitable.

Figure 1 illustrates the same data, where the black solid and dashed line represent the costs for the Google Cloud and on-premises solution, respectively, over time. The shadowed area around both lines indicate a calculated range of variance for a mean usage between 10% and 70%. Additionally, for both variants two further lines, indicating the theoretical trend for 20% and 40%, are indicated by the gray lines.

More specifically, Figure 1 renders the trade-off between cloud and on-premises computing in terms of costs. The on-premises solution faces the issue of high costs in the initial phase, but far less further costs. Using cloud services one pays exactly what is consumed. Depending on the actual usage, on-premises can pay-off sooner or later in case the system set-up may allow to save expenses during procurement or the running ongoing costs.

In literature also other comparisons on the costs of cloud vs. on-premises computing can be found, see [6].

B. GDPR

The protection of the intellectual properties and values of an institution or conforming to GDPR can be one of the reasons which force professionals to refrain from using cloud resources. Although, the large established cloud providers (e.g. Microsoft, IBM, Amazon, Google) provide possibilities to rent services which are hosted in particular regions, it is important to note that those are mainly American companies which may be forced to provide information, possibly sensitive data, even if stored in some other a non-us region.

C. Other issues with Cloud Resources

In the following other important issues when dealing with cloud resources are listed:

- Attacks: cloud resources might be more vulnerable to attacks, not because they are more vulnerable in principle but with services, domains and exploits well-known by hackers they are rather under attack than custom services unknown to the (international) public.
- Network connectivity dependency and downtimes: Challenges arising due to technical outages of the world wide web may severely brake your manufacturing e.g. when thinking about production processes demanding real-time processing.
- Limited control and flexibility: Although, cloud providers getting richer in terms of tools and services, controlling remote hardware, servers or services may be still quite hard to deal with and may come with their own peculiarities which are difficult to manage.
- Vendor lock-in: provided services and interfaces are often highly proprietary and do not conform to cloud native structures or even basic IT standards.
- Disaster Assistance and Data Backup and Recovery may grow in complexity if e.g. a non-cloud based backup is required for cloud resources. However, for the GPU cluster backup only exists in form of a recovery image which restores the initial installation state. Further, as this cluster is used as a computing engine, which is doing batch processing, data on that cluster also resides on other systems and is not lost when data corruption occurs.

III. Requirements to our on-premises infrastructure

The intended infrastructure is used for different purposes (research, course work) and by different groups of people (faculty, scientific staff, and students). Each of the stakeholders has different requirements for the setup. Before the design of the architecture, the requirements were defined in order to select the components of the software stack and the topology of the cluster accordingly. For our setup, we identified requirements, which are discussed in the following.

(A) **Ease of usage** Development of all algorithms and jobs to be executed is done on local machines. To execute the jobs on the cluster, no modifications on code and no major changes on the call are necessary. Data and other necessary resources need to be available for the job, where ever it is scheduled to be executed. Additionally, no knowledge about the system itself is necessary for execution.

(B) **Scheduling of Jobs** A jobs execution is decoupled from the scheduling. This means, that a users, who submits a job, does not have to know about the architecture of the cluster or the setup of the computing nodes. When submitting a task, this is appended to a queue of jobs. When executing a job, the request for a certain type of resource is stated (e.g. type of GPU, memory, etc.). The system schedules the jobs according to first-in-first-out and the availability of the requested resources.

(C) **Workload Distribution** Resources shall be used equally over the system, where possible. If several computing nodes with likewise configuration exist, it must be assured that the load of work is distributed among those nodes. It must be avoided, that only few components take care of the whole computation, whereas the others are not used.

(D) **Permission Management** Different stakeholders use the cluster for different purposes. Hence it shall be possible to assign users to groups to restrict their usage to a defined policy. As a figurative example, members of the group *students* shall only be allowed to use x GPU at a time for a maximum of n hours. This shall reduce the potential risk of misusing the infrastructure.

(E) **Maintainability and Scalability** It shall be possible to remove and exchange parts of the system, without too much time effort, and downtime. The same applies to the extension of the system: For future developments, the system must be designed in such a way, that additional computing power can easily be extended with no major modification to the system itself, which enhances complexity to the architecture and the configuration. Heterogeneous setups should be configurable.

(F) **Network Speed** When designing the topology of the connection between all the components within the system, it shall be taken care of to design this system in such a way, that the network speed is high and influences from other network traffic that reduce speed kept low. On the other hand, also the traffic between the components must not have any influence on the other resources on the network. Printing, access to the web or file servers must not be reduced in speed, just because of transferring data among two nodes.

(G) **Costs** The last requirement concerns the monthly costs of the infrastructure. Monthly costs are intended to be as low as possible, while initial costs must be below a certain predefined budget.

IV. Infrastructure Architecture

During the research, several solutions, offered by various companies were found, like [7]. All of these solutions posed the drawback to exceed the defined procurement budget.

The architecture of the up-scaled infrastructure is motivated by [8] and inspired by [7]. The designed architecture is illustrated in fig. 2. The whole cluster is completely abstracted from the public network by an interface router. This interface router creates a private network for the whole infrastructure, without putting load on the public network by data synchronization between the nodes. As a result, this hides the complexity of the infrastructure to the outside, while being still accessible from the public network. For the implemented cluster, a *MikroTik CRS-309-1G* is being used.

For the private, cluster network, instead of a conventional *Gigabit Ethernet*, a *SFP+* connection is established. This protocol support data rates up to 10 Gbit/s. This enables faster synchronization between the nodes.

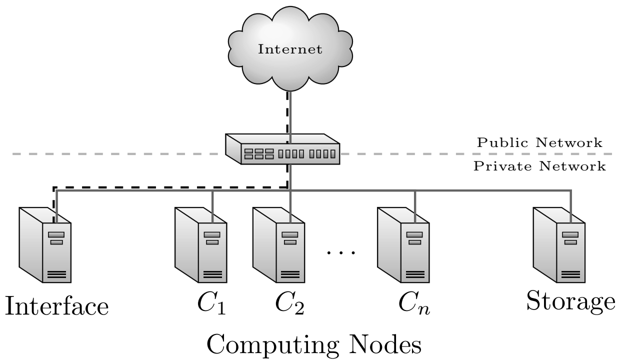


Fig. 2. The network architecture of the setup contains an interface node, several computing nodes ($C_1 \ldots C_n$), and a storage node, all hidden behind a router. The dashed lined indicates the default route for the ssh connection.

Within this private network, several nodes are connected. The interface node poses as the entry point to the cluster and the only accessible machine within the network for users. On this device, all tasks are scheduled and distributed according to the set policy and configuration. This node does not have any GPU, so no accidental blocking of resources may happen. For the setup, the interface node is the standard gateway for the *SSH* access.

The actual computing tasks are scheduled to the computing nodes ($C_1 \ldots C_n$). These machines are equipped with sufficient computing power, GPUs, and memory. For the implementation, the already available machine (old infrastructure) C_1 is being used. Additionally, a further machine C_2 is added to the system. The configuration of all machines (C_1, C_2, and I) are described in Section IV.

As indicated, this architecture also makes it easy to increase the number of further nodes. In fig. 2, a dedicated storage node is indicated. This storage is available by all components within the cluster and enables access to all data by all nodes. For the current implementation, this storage is not included in the setup. A dedicated storage will be added with one of the future extensions. Currently, only the storage from the nodes C_1 and C_2 is used by all nodes.

All devices and nodes in the implemented setup are chosen, so that they are easily built into a existing server rack within the in-house server room.

V. Cluster Setup

For the setup of the cluster and the software stack, extensive research was needed. This was caused by the variety of software products available and their interoperability, enterprise products on the market, and expensive reference solutions. The following setup is inspired by the huge computing infrastructure of the University of Massachusetts [9]. The selection and installation of the software stack is based on the instructions in [10].

Each node in the initial setup of the cluster is based on a long-term support (LTS) Linux distribution. In the setup a *Ubuntu 20.04 LTS Server* is used. The decision towards a LTS

TABLE II
CONFIGURATION OF THE NODES AS IMPLEMENTED WITHIN THE SETUP.

Node	**GPUs**	**CPUs**	**RAM**	**Local Storage**
I	-	1x AMD EPYC 7302P, 3.00 GHz	126 GB	460 GB system
C_1	8 NVIDIA GeForce RTX 2080Ti	2x Intel Xeon Silver 4114, 2.20 GHz	230 GB	230 GB system + 8 TB data
C_2	3 NVIDIA RTX A6000	2x AMD EPYC 7452, 2.35GHz	512GB	240 GB system + 8 TB data

version was done for a long lifetime support. Among others, consumer options used in practice are *CentOS Linux 7* [9], and *Ubuntu 20.04 LTS* [10]. A comprehensive overview and comparison of the operating systems and components used in other clusters can be found on the *TOP 500*[6] list.

On each of the computing nodes, the appropriate CUDA driver is installed. The correct driver and CUDA version is dependent on the GPU and operating system.

Since the jobs to be executed are scheduled by a central entity on a machine within the cluster, storage synchronization is crucial. A user is only able to access the interface node. All data and environments also need to be synchronized to the other nodes, so that the execution of the scheduled jobs work properly. As noted in Section IV, no dedicated storage node is implemented within the cluster. The storage integrated in both computing nodes, C_1 and C_2, is provided as storage for the cluster. The storage is used as a ZFS (Zetta file system) and mounted from all other nodes within the cluster. Each machine mounts that storage to the same mount point in the file system. With other setups different approaches can be found, where either everything is mounted using only one directory [10] or the storage split among several directories for different purposes (e.g. home directories, research storage, scratch space, temporary Space) [11]. Within the implemented cluster, only the home directories and a data directory are shared within the cluster.

For password-less connections from the master to all worker nodes, the ssh keys are exchanged and munge is used as an authentication service. For the application of the used workload manager, MariaDB is used.

As a workload manager, the decision was made towards SLURM [12]. Another workload managers commonly used is HTCondor [13]. The decision towards SLURM is not only based on the reference implementations, but also since SLURM is easy to use. Let's consider a simple script, name `execute_me.py`. Using SLURM, applying a standard configuration, it is scheduled as simple as `sbatch python execute_me.py`. No further adaptations are needed to be taken care of.

Additionally, SLURM, which has a large community, is well documented, and even has enterprise support. Furthermore, over 60% of all supercomputers use SLURM in their setup [14]. SLURM can be used for any size of clusters and works well with over $1,200$, partially different, GPUs [9] and to smaller setups and clusters. Both, HTCondor and SLURM, can be used interoperable. An overview on this and on both tools itself is given by [15] and [16].

SLURM is a highly configurable component. This workload manager can be configured in the most basic approach, where it works only as a simple workload manager that balances jobs on all nodes, up to a system were it is a part of a software stack, that not only restricts access to resources for certain users but also interconnects with other plugins like for accounting for the actual usage. For the implemented system, the initial configuration has two groups of users, where different types of GPUs are restricted to certain user groups. For instance, users of the group *factuly* may use all GPU, and members of *students* are restricted to GPUs on computing node C_1. Additionally, users from the first group can schedule as much jobs as needed, where the jobs are executed as soon as possible, and members of the latter can only have one job running at a time.

Users and groups are managed using *FreeIPA*[7]. This software component offers an intuitive Web-UI for managing all accounts. For the initial setup, the clusters user accounts are intentionally independent of the organizational accounts for the rest of the official IT infrastructure. *FreeIPA* allows not only to add and delete users and groups, but it also allows to set validity periods, storage quotas, and much more.

VI. FUTURE ADAPTIONS

The purpose of this cluster is in computational power for research and academia. In both fields, it will be likely that the demand for computational power will increase in the following years. For this purpose, several adaptions to the cluster are planned already.

The first extensions on the number of computation nodes will be with a more 'low-spec' hardware. Using already available consumer hardware, the cluster will be extended, so that the number of computational devices increases. Using, e.g. several discarded, slower hardware can be offered in exchange for more jobs in parallel. Given the current situation on the international market, where only a small number and on overpriced GPUs are available, this is a very attractive option as long as the memory requirements of jobs are not too high. Additionally, from an economical perspective, this gives the hardware a second life instead of being recycled.

Depending on future usage, also more memory computational tasks may be executed on the machine. Therefore, SLURM also needs to be configured, so that no GPU, but a certain amount of memory can be allocated.

As noted in Section V, no dedicated storage is used within the implemented cluster. A further extension, where dedicated

[6] https://www.top500.org/

[7] https://www.freeipa.org

storage is aimed. This storage is also planned to be fault-tolerant, by using formats of storage virtualization.

Maintaining a continuously growing infrastructure, it is not easy to keep an overview of the status on all machines and devices in the cluster simultaneously. To make administration easier, a cluster management system will be implemented. Observing the status of all connected nodes and devices, managing them, and taking actions on events will be the main task of the component implemented by this extension.

In Section II, the comparison of the costs of using the cloud infrastructure is elaborated. Depending on the usage of the cluster, it is a further use case to overcome performance peaks by including cloud resources by an external vendor/provider. By starting a cloud instance while the cluster is in high demand, quick jobs can be outsourced to the cloud, which speeds up the execution of jobs.

Depended on future usage, it is also possible to improve the hardware to enable computational jobs, with a memory demand higher than the memory of a machine. RDMA[8] is an extension technology with the ability to access memory from one host to another remote.

VII. Discussion and Conclusion

In this work, we focused on an overview of the transformation of a research machine with several GPUs to an extensible cluster of several computing nodes, each offering several GPUs.

The requirement on the ease of usage (requirement (req.) (A)) is preserved. Due to the usage of SLURM, no modifications need to be done on the code or the call itself, only the execution needs to be done using SLURM. Also, the scheduling of jobs (req. (B)) and the workload distribution (req. (C)) are handled by SLURM. The permission management (req. (D)) is also covered, commonly by SLURM and *FreeIPA*.

The requirements on maintainability and scalability (req. (E)), as well as on network speed (req. (F)) are both covered by the design of the clusters architecture. Further machines can be easily added with a minimum configuration effort. So even when starting with a small number of computing nodes, like in our case with two, it exhibits an effective and efficient platform to work with. The maximum speed within the network is above the standard network used within our institution and the traffic on the cluster does not influence the public network, due to its private scope.

The costs (req. (G)) are kept low and within budget. One might argue, that the components in this proposed setup are still not from a low-cost price range. However, all components described within this work can be replaced with existing and more budget-friendly components. Additionally, the monthly costs are controlled, without surprises by an unexpected increased demand.

In the introduction, we posed four questions that were asked before this project and answered within this work. Q1 was concerned, whether it is beneficial for the proposed use case to use cloud resources. In Section II, this topic is elaborated extensively. By showing a comparison of the costs, the decision was made against cloud solutions and to foster an on-premises infrastructure. Also, preexisting solutions (Q2) were not considered, since our research showed, that those solutions are above the given budgets limit. The design of the architecture (Q3) and the components of the software stack (Q4) are shown in Section IV and Section V, respectively.

In growing enterprises, data science departments, and research facilities data scientists, often the only computer science or engineering experts, also have to take care of their infrastructure. A simple solution is established fast, but scaling is hard. A lack of expertise often results in sub-optimal or redundant solutions or endless research.

In this work, we showed from a data scientist perspective, how to scale an existing infrastructure with a limited budget, while, among other requirements, maintaining extensibility. We presented our architecture and the software stack of our cluster with job scheduling.

[8] Remote Direct Memory Access

References

[1] S. Hooker, "The Hardware Lottery," *arXiv:1911.05248 [cs]*, 2020. [Online]. Available: https://arxiv.org/abs/1911.05248

[2] M. M. Waldrop, "The chips are down for Moore's law," *Nature*, vol. 530, no. 7589, pp. 144–147, Feb. 2016.

[3] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015, arXiv: 1409.1556.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[5] J. Baschab and J. Piot, *The executive's guide to information technology, Second Edition*. John Wiley & Sons, 2007.

[6] C. Fisher *et al.*, "Cloud versus on-premise computing," *American Journal of Industrial and Business Management*, vol. 8, no. 09, p. 1991, 2018.

[7] D. Amette, S. Ranganathan, A. Borulkar, S.-H. Lin, and S. Rao, "Scalable AI Infrastructure: Designing for Real-World Deep Learning Use Cases," NetApp, White Paper NVA-1121, Mar. 2019.

[8] P. Gupta, *How to Build a GPU-Accelerated Research Cluster*. NVIDIA, Apr. 2013, nVIDIA Developer Blog. [Online]. Available: https://developer.nvidia.com/blog/how-build-gpu-accelerated-research-cluster/

[9] U. of Massachusetts, "Gypsum Cluster Documentation." [Online]. Available: https://gypsum-docs.cs.umass.edu

[10] N. George, "slurm_gpu_ubuntu," Nov. 2020. [Online]. Available: https://github.com/nateGeorge/slurm_gpu_ubuntu

[11] "Description of the GPU Cluster," Jul. 2021. [Online]. Available: https://hpcf.umbc.edu/

[12] A. B. Yoo, M. A. Jette, and M. Grondona, "SLURM: Simple Linux Utility for Resource Management," in *Job Scheduling Strategies for Parallel Processing*, G. Goos, J. Hartmanis, J. van Leeuwen, D. Feitelson, L. Rudolph, and U. Schwiegelshohn, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, vol. 2862, pp. 44–60, series Title: Lecture Notes in Computer Science.

[13] "Computing with HTCondor," Jun. 2021. [Online]. Available: http://www.cs.wisc.edu/condor

[14] Y. Robert *et al.*, "TOP500," in *Encyclopedia of Parallel Computing*, D. Padua, Ed. Boston, MA: Springer US, 2011, pp. 2055–2057.

[15] C. Hollowell, J. Barnett, C. Caramarcu, W. Strecker-Kellogg, A. Wong, and A. Zaytsev, "Mixing HTC and HPC Workloads with HTCondor and Slurm," *J. Phys.: Conf. Ser.*, vol. 898, p. 082014, Oct. 2017.

[16] R. Du, J. Shi, J. Zou, X. Jiang, Z. Sun, and G. Chen, "A Feasibility Study on workload integration between HT-Condor and Slurm Clusters," *EPJ Web Conf.*, vol. 214, p. 08004, 2019.

Strategic Approaches to the Use of Data Science in SMEs

Lessons Learned from a Regional Multi-Case Study

Cornelia Ferner and Thomas Heistracher
Salzburg University of Applied Sciences, 5412 Puch, Austria
{cornelia.ferner, thomas.heistracher}@fh-salzburg.ac.at

***Abstract*—The potential of data analytics and modeling, especially in the context of digital transformation, is well known and seen as an opportunity to increase competitiveness and innovation in companies. However, the technical and organizational implementation of these methods poses challenges specifically for SMEs. A regional multi-case study explores the causes and exhibits patterns that suggest differentiated recommendations for action. Awareness of the diverse limitations and challenges can help all players involved – companies, universities and governmental organizations – to design future projects in an even more targeted manner.**

***Index Terms*—data science, digital transformation, SME, triple helix**

I. Introduction

Data science is a team effort that usually requires several roles: A data scientist that is not only proficient in mathematics and programming, but can also apply machine learning methods to target applications. A data engineer, who is responsible for the physical data storage, which includes infrastructure design, data access, data security etc. A data analyst that analyzes existing data and creates visualizations, reports or dashboards. Higher-level management tasks are usually taken over by a Chief Data Officer (or Data Manager), who reports to the executive management [1].

While big companies have increasingly large data science teams, SMEs most often do not have comparable capacities. Fortunately, this does not need to keep them from developing innovative ideas and implementing data-centric projects. Innovation and economic development is often driven by the successful collaboration of three players: academia (universities), governmental organizations and the industry. This interplay is known as Triple Helix Model [2], that formalizes the interdependencies and interactions between the three players responsible for a region's long-term technological development.

Exchange with and input from universities and other research facilities can help with the realization of new ideas. University-industry collaborations provide access to additional know-how and human resources outside the company and often enable testing of new ideas without risk to the production environment. Companies cannot flourish without the infrastructure and legal framework provided by the government. Besides legal frameworks for copyrights, taxation and location policy, the government's role is to set funding programs to encourage collaboration and to provide incentives for innovation. The interplay between government and academia defines the long-term strategic research direction and can shape both the academic landscape as well as the job market.

The federal state of Salzburg offers a specific funding program designed to support the digitization efforts of regional companies, mainly SMEs. In order to assess the program, the currently funded projects in the context of data science were evaluated. The companies that participated in the multi-case study were selected with the support of the regional innovation service agency, taking care to cover as diverse a spectrum of industries, company sizes within the category SME and geographic locations within the federal state of Salzburg as possible; the ICT sector and e-commerce businesses were deliberately excluded here.

As a result, twelve semi-structured interviews [3] were conducted via video calls with experts from the companies who were directly involved in the project implementation or who played a leading role in the strategic development (e.g. owners, (technical) managers etc.). The guideline for the interviews included closed questions that are in line with the levels and dimensions of the Data Science Maturity Model (see Section II) and qualitative questions to capture the companies' self-assessment from their own perspectives, especially regarding risks and future opportunities that came from their projects.

The contribution of this paper is two-fold: First, we present the findings of our multi-case study, where we analyze regional SMEs based on a simplified data science maturity model and highlight interesting insights. Second, we derive

The study presented in this publication was conducted as part of the research project "DataKMU – Vernetzung und Wissenstransfer im Bereich Data Science" that is funded by the European Regional Development Fund with grant number AB215.

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_12

recommendations for the three categroies of players involved to even increase future project output and foster innovative collaboration.

II. Data Science Maturity Model

The Data Science Maturity Model [4], introduced by Mark Hornick, Senior Director at Oracle Data Science and Machine Learning, allows to assess a company's current state concerning data science strategy and technology:

"Enterprises that already embrace data science as a core competency, as well as those just getting started, often seek a roadmap for improving that competency. A data science maturity model is one way of assessing an enterprise and guiding the quest for data science nirvana. Upping an enterprise's level of data science maturity enables extracting greater value from data for making better data-driven decisions, realizing business objectives more efficiently, and having a more agile response to changing market conditions." [5], p.3.

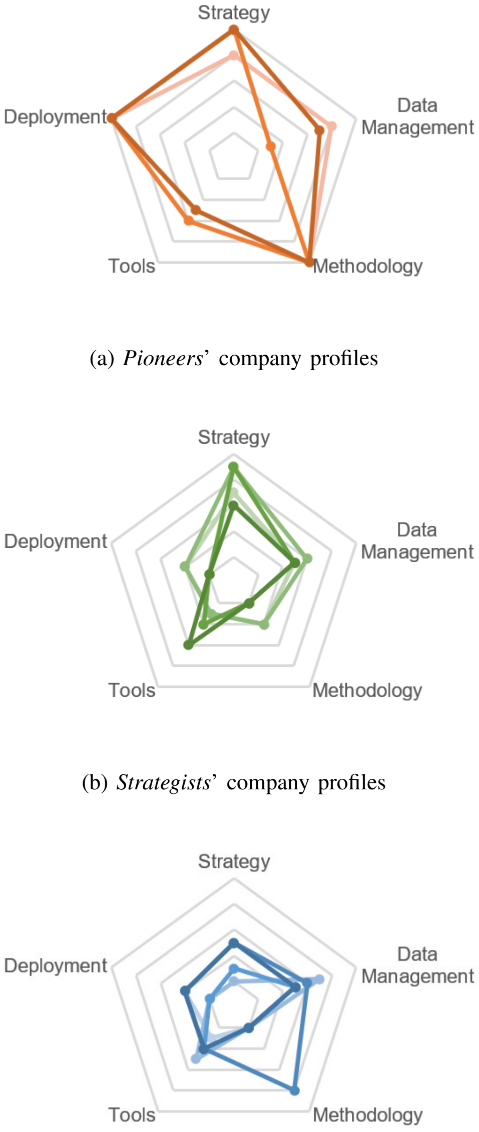
The data science maturity model identifies ten dimensions that are decisive for the success of data science projects. Each dimension consists of five maturity levels - from basic level 1 to the most mature level 5. Each company can individually specify at which level they are or intend to be in each dimension.

Given that the surveyed companies do not have dedicated data science teams, their profiles where assessed in five dimensions: *strategy*, *data management*, *methodology*, *tools* and *deployment*. The dimension *strategy* captures the strategic orientation of a company with regard to data use and exploitation and assesses, whether data is regarded as a by-product or as capital. The dimension *data management* captures the type of data storage that is used – central or distributed systems locally in the company or outsourced to external providers. The *methodology* dimension captures how companies use methods to analyze the past and forecast the future. Within the dimension *tools*, the usage and scalability of software packages is monitored. The fifth dimension, *deployment*, describes how results are reported, from static files to dashboards to continuous deployment of dynamic models.

III. Multi-Case Study: Data Science in SMEs in Salzburg

Even if not every company needs to develop a data-centric business model [6], the current focus is on driving digitization, at least in certain company areas. Recording and using data alone is not sufficient. The focus needs to shift towards exploiting the data value by creating new insights and deriving action items. In this context, we talk about data science [7], i.e. the analysis and modeling of data.

In our multi-case study, the experiences and findings of twelve SMEs in Salzburg concerning their implementation of data science projects were surveyed. In the absence of in-house IT and software development teams, let alone dedicated teams for data science, digitization projects in these companies are fundamentally collaborative efforts. Only if



(a) *Pioneers*' company profiles

(b) *Strategists*' company profiles

(c) *Pragmatists*' company profiles

Fig. 1: Different company profiles with regard to the five dimensions of the Data Science Maturity Model. The three groups were assigned through clustering. The different color intensity is used to discriminate the individual company profiles.

Pioneers	Legal questions with respect to machine learning
	Infrastructure, equipment
	Acquisition costs
	Application-specific challenges

(a) *Pioneers*’ challenges

Strategists	Missing manpower and/or lack of specific training
	Cost-benefit analysis
	Costly data management
	Extensive data maintenance

(b) *Strategists*’ challenges

Pragmatists	Data transparency and security
	Low data volumes (”small data”)
	Lacking interfaces to current software
	Lacking overview of methodologies and available software tools

(c) *Pragmatists*’ challenges

TABLE I: Main challenges in data science projects mentioned by the three different company groups.

motivated employees get involved beyond their actual duties and are open to new ideas such projects can be realized at all.

When asked about the motives for implementing data analytics projects, intrinsic motives predominated in the interviews. The desire for further development and technological progress, the expansion of the product portfolio or services offered, as well as resource optimization (time, costs) were named as reasons. In addition, legal and certification requirements or explicit customer wishes are also drivers of digitization.

Systematic analysis of individual company profiles with regard to the five data science maturity dimensions reveals patterns, as shown in Figure 1: A group of companies, which we refer to as the *Pioneers*, achieve the highest level of proficiency in some dimensions and act in a data-centric manner with regard to strategy, management and implementation (see Figure 1a). Another group of companies shows a peak at the *strategy* dimension (see Figure 1b), indicating that these companies are aware of the value of their data and the benefits they can derive from it. We refer to this group as *Strategists*. The profiles of the last group of companies does not show this peak (see Figure 1c). The strategic use of data plays a less important role for them than the actual implementation. This third group is referred to as *Pragmatists*.

The clear differences between the company profiles that allows a grouping into three distinct clusters rises the question, which underlying factor is common to the companies within each group? Which factors determine the different approaches towards data science projects? Reasons such as company size or culture, or a common industry sector can be ruled out, as those differ within each group anyway.

An explanation for the different maturity profiles is revealed by the answers the companies provided to the question about the main challenges when implementing data science projects (see Table I). While those challenges are very application-oriented among the *Pioneers* (see Table Ia), decisive differences can be identified between the *Strategists* and *Pragmatists*:

The *Strategists* are most likely to see the availability of know-how and trained employees as a challenge (see Table Ib). With the implementation of data-driven projects, the demands with regard to employees would increase, which would require different training or further education. In addition, companies from the *Strategists* group mention the difficulty of recruiting new employees with the appropriate training, as salaries comparable to the IT industry would often exceed their own salary structure. The *Strategists* also draw attention to the need to weigh up the costs and benefits before implementing data-driven projects, which also underscores the clear peak in the strategy dimension. The cost-benefit consideration remains an issue even after successful implementation, when it comes to the effort required for continuous data management and maintenance.

The *Pragmatists*, on the other hand, cite legal considerations such as data transparency and security as a fundamental challenge. When implementing data science projects, they find the provision of interfaces and subsequently the connection of data science tools to existing software to be challenging. In this context, the lack of comparability and confusion of available tools is also addressed (see Table Ic).

The differences between the *Pragmatists* and the *Strategists* result from the type of data the respective companies work with. Companies with a clear strategy for how to use the data own that required data. They are aware of the available volume and quality and can make an initial assessment of what information can be generated. Companies in the *Pragmatists* group are largely dependent on the use of external data they do not own themselves. The availability or possibility of use as well as the data quality are comparatively uncertain or unknown. Table II lists examples of different types of data according to their source and the volume to be expected.

IV. Results

Data Science and Machine Learning methods that are applicable in a project depend on the type of data and the available volume. In addition to fundamental data protection issues, the primary challenge especially in the case of external data is the merging of heterogeneous sources. Missing interfaces and a common data format need to be implemented. In some cases, data must be made accessible step by step from external sources or alternatively generated by suitable methods within the company itself. This problem is referred to as cold start problem: Many methods for data science require large amounts of data to be successful in automated analysis and modeling. If this data is not (yet) available in sufficient quantity, simpler methods can be applied in the short term or data can be simulated as a workaround.

		Source	
Volume		internal	external
	high	Plant data Machine data Control signals Measuring signals Process data Log data	Network data Access data Utilization data Log data Energy consumption data
	low	Employee data Customer data Order data	Data from customers Data from suppliers Data about products, parts Data about (raw) materials

TABLE II: Matrix with different types of data according to their source and volume to be expected.

A different challenge is posed by a high number of internal data sources: Machine and process data in particular are often available in very high resolution (e.g. every millisecond) and thus in large quantities. This raises the question of suitable data reduction methods either by pre-filtering or aggregation. Applicable methods mainly depend on the amount of data to be processed or on the type of interfaces required.

For each project and each task, it is therefore necessary to assess the ”data inventory” and its requirements in order to be able to develop customized solutions. A pure blueprint implementation is only possible in the rarest of cases. This is why domain expertise with extensive knowledge of the (business) processes that generate the data is needed, as well as data science experts with the skills to select suitable methods and adapt them to the specific needs [8].

From these insights, we derive recommendations for each of the players of the triple helix:

A. SMEs

Companies can assess their current data science maturity level based on the model presented in Section II. The resulting profile allows to align the future strategy and goals. In addition to the maturity profile, a ”data inventory” creates awareness of internally available data and possible external dependencies, which sets the technological and methodological frame for future projects. As more and more jobs require working with data, it is recommended to promote data literacy among all employees in the company.

B. Universities

Data science projects are optimal starting points for industry-university collaborations: SMEs with limited trained staff in programming and/or data science can benefit from the external technological know-how of academic reasearchers and can contribute the required domain expertise to scientific research projects. In addition, collaborations with universities provide access to an extended pool of personnel.

From a university's point of view, a data science project allows for interdisciplinary collaboration, as applied data scientists can team up with business informatic scientists and machine learning researchers.

C. Government

With their digitization program, the government of the federal state of Salzburg has filled a funding gap by especially targeting regional SMEs that take their first steps towards more digitization. While many companies have already benefited, the results from our structured interviews show another potential: An additional benefit can arise from extending the programs to also grant funds to companies owning data, but not actively developing data science projects. The provision of data (e.g. by suppliers, manufacturers, producers) can help other companies to get access to ”missing links” in their projects through ”data partnerships” [8]. Allowing such companies to jointly apply for funding would help in making more data (publicly) available and framing data as a business case. This could also serve to bring about a cultural change towards open data in the realm of data science.

V. Conclusion

As part of a survey of selected companies in the province of Salzburg, the importance of the topics of digitization and data science was surveyed, especially for SMEs in non-ICT industries. All companies recognize data science as an added value for their business models and have gained initial experience in implementing data science projects. The evaluation of the interviews shows a differentiated picture regarding the companies' approach to these projects that is partly caused by the availability and volume of the required data. As the appropriate methods and tools for generating data insights and predictions depend largely on the type of data, smaller explorative studies can help in evaluating approaches and assessing their effectiveness systematically without risk. By applying for advanced funding programs, university partners can also be involved in research projects that step-by-step lead to higher data science maturity levels.

References

[1] T. Stobierski, “How to Structure Your Data Analytics Team,” *Harvard Business School Online*, 2021, available online: https://tinyurl.com/rtmaxvk.

[2] H. Etzkowitz, *The Triple Helix: University-Industry-Government Innovation in Action*. New York, NY: Routledge, 2008.

[3] A. Bayman and E. Bell, *Business Research Methods*. Oxford: Oxford University Press, 2011.

[4] M. Hornick, “Data Science Maturity Model - Summary Table for Enterprise Assessment,” Oracle R Technologies Blog, 2018, available online: https://blogs.oracle.com/machinelearning/post/data-science-maturity-model-summary-table-for-enterprise-assessment-part-12.

[5] M. Hornick, “Data Science Maturity Model for Enterprise Assessment,” Oracle, Tech. Rep., 2020, available online: https://www.oracle.com/a/devo/docs/data-science-maturity-model.pdf.

[6] S. Andriole, “Five Myths about Digital Transformation,” *MIT Sloan Management Review*, vol. 3, no. 58, pp. 20–22, 2017.

[7] J. Kelleher and B. Tierney, *Data Science*. Cambridge: The MIT Press, 2018.

[8] U. Kruhse-Lehtonen and D. Hofmann, “How to Define and Execute Your Data and AI Strategy,” *Harvard Data Science Review*, 7 2020, available online: https://hdsr.mitpress.mit.edu/pub/4vlrf0x2.

Minimal-Configuration Anomaly Detection for IIoT Sensors

A Systematic Analysis of Deep Learning Approaches

Clemens Heistracher*, Anahid Jalali*, Axel Suendermann†,
Sebastian Meixner†, Daniel Schall†, Bernhard Haslhofer* and Jana Kemnitz†
*Austrian Institute of Technology, 1210 Vienna, Austria
†Siemens Technology Austria, 1210 Vienna, Austria
{clemens.heistracher, anahid.jalali, bernhard.haslhofer}@ait.ac.at,
{axel.suendermann, sebastian.a.meixner, daniel.schall, jana.kemnitz}@siemens.com

***Abstract*—The increasing deployment of low-cost IoT sensor platforms in industry boosts the demand for anomaly detection solutions that fulfill two key requirements: minimal configuration effort and easy transferability across equipment. Recent advances in deep learning, especially long-short-term memory (LSTM) and autoencoders, offer promising methods for detecting anomalies in sensor data recordings. We compared autoencoders with various architectures such as deep neural networks (DNN), LSTMs and convolutional neural networks (CNN) using a simple benchmark dataset, which we generated by operating a peristaltic pump under various operating conditions and inducing anomalies manually. Our preliminary results indicate that a single model can detect anomalies under various operating conditions on a four-dimensional data set without any specific feature engineering for each operating condition. We consider this work as being the first step towards a generic anomaly detection method, which is applicable for a wide range of industrial equipment.**

***Index Terms*—Internet of Things (IoT), Industry and Production 4.0, Predictive Maintenance, Unsupervised Machine Learning, Anomaly Detection**

I. Introduction

Prognostics and health management approaches have been studied extensively across industrial applications, such as aircraft engines [1], wind turbines [2], and other expensive and mission critical machines. The application of IIoT sensors [3]–[5] enables continuous monitoring on previously unequipped industrial assets. The SITRANS multi sensor [5] enables affordable to monitor equipment and industry expects huge cost savings from the implementation of data-driven predictive maintenance techniques such as anomaly detection. However, the implementation of a traditional anomaly detection technique for some specific equipment often requires significant manual feature engineering [6], [7] and model optimization effort. Alternative approaches requiring less effort and transferability to similar equipment are therefore becoming increasingly relevant in industrial contexts. A number of anomaly detection methods for predictive maintenance have been proposed. Kato et al. [8] proposed a rule-based approach for fault detection in spacecrafts. Principal component analysis (PCA) based anomaly detection in networks was discussed by [9]. [10] showed that autoencoders can outperform PCA based approaches for telemetry data of spacecrafts. Unsupervised anomaly detection with deep autoencoders was shown by [11], [12].

Our aim is to develop an unsupervised anomaly detection method for a universally deployable IIoT sensor tag, which records multivariate data. It should learn anomalies automatically over time and thereby reduce manual feature engineering effort. Our specific contributions so far are: (i) define initial requirements and derive design rationals for minimal-configuration anomaly detection for IIoT Sensors (ii) provide a hand-crafted benchmark data set of evaluating anomaly detection approaches, and (iii) train various deep neural networks with autoencoder architectures and evaluated them against benchmark models.

II. Industrial Requirements and Design Rational

A low-cost multi sensor should be applicable to any industrial asset. With the help of this multi- sensor, the conditions should be monitored without any meta information available. For this purpose, the healthy state with all typical operational conditions is recorded with and used as a reference for anomaly detection. The amount of healthy training data is not strictly limited and several days can be expected. The system should be minimal configurable. The only input parameter is the healthy reference data. The system should be operable by a non-machine learning expert. The user should be a domain expert and select a time period with typical operational conditions as reference. Everything else is left to the model and the system. The decision for an unsupervised machine learning paradigm results from the requirements.

III. Data Set Creation

We have selected the peristaltic pump as it can be operated using various operating conditions and anomalies in the pipe's water flow resistance can be applied easily. Further, the rotor of a peristaltic pump is representative for many rotary equipment in the industry, such as fans, compressors and turbines. Additionally, degradation is commonly observed and the can be controlled as pipes can be replaced easily. Abrasion of the pipes will be used to predict failure of the pipe system

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_13

in subsequent work. Peristaltic pumps are used for sterile or aggressive liquids, as the pump doesn't contact the fluid. The flow of liquids is induced by a repeating sequential compression of a flexible tube that pushed the liquid in one direction.

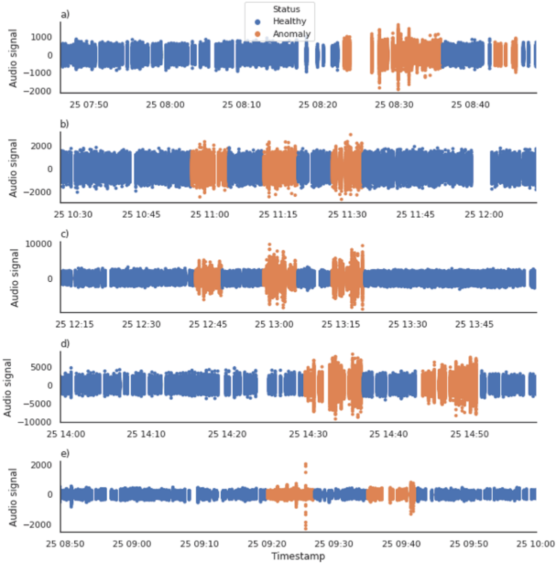


Fig. 1. The audio signal for various operating conditions: a) 100 Hz, b) 150 Hz, c) 200 Hz, d) 250 Hz, e) 50 Hz with an additional 12 hours of normal state.

We created our data set with a prototype of the SIEMENS SITRANS multi sensor specifically developed for industrial applications and harsh environments [5]. The sensor offers a multiple number of measurement parameters. In this work, we used the sensor tag that records three-axis vibration, each with a frequency of 6664 Hz , an audio signal with $16k$ Hz, and temperature. Due to restrictions in bandwidth, vibration and audio are measured sequentially for 1024 data points every 60 s. We mounted the sensor tag on at rotational axis of the pump's rotor and documented the sensor's angle and its horizontal axis. To simulate various operating conditions, we operated the pump under various conditions by changing the pump's frequency. Further, we induced anomalies by restricting the water flow in the tube leading to the pipe. We scheduled the data acquisition to generate a data set that is balanced for operating conditions and anomalies. Additionally, we documented the replacement of the tube to allow analysis of the tube's degradation and we also perform measurements with a rotated sensor to evaluate the models robustness against rotation. A model, which performs well if the sensor was rotated and reattached, is a candidate for architectures that are easily transferable across equipment and require minimal configuration effort. The data set contains 3041 samples with each 1024 data points for audio and the three vibration axis and will be made publicly available.

IV. Experimental Setup

We trained unsupervised machine learning models to detect the anomalies in our dataset by using autoencoders based on a fully connected deep neural network (DNN), long short-term memory (LSTM) networks and convolutional neural networks (CNN). Autoencoder networks are trained to reproduce a input signal by minimization the error between input and output signal, which is called the reconstruction error. This is done by setting the input values as the target values. If there is a layer with a feature space lower than the input space, the autoencoder is forced to learn a compressed representation and therefore needs to generalize and approximate the input. In other words, a bottleneck in the network requires the encoder to extract the most substantial information.

For anomaly detection, autoencoders are trained to reconstruct only healthy machine data. It is assumed that the autoencoder learns to reconstruct the input for healthy machine data, as it was trained to do so, but will fail to reconstruct anomaly data. The reconstruction error — the error between input and output signal — can be used as an anomaly score. A reconstruction error above a threshold indicates an anomaly. The threshold is calculated on a subset of the healthy data that was excluded from training, by calculation mean + standard deviation of the reconstruction error on the subset. We evaluated the effectiveness of the anomaly detection using standard accuracy (Ac.), precision (P), recall (R) and the F1-score.

We compared the performance of our models on a variety of features. We used the audio and the vibration (vib. 3D) signal separately as well as a combinations of both. Further, we use raw signal and the fast Fourier transform (FFT) of the signals. In order to achieve invariance towards rotation of the sensor, we also use the euclidean norm of the three-dimensional vibration signal, which is denoted as vib. 1D. Feature names are generated from the options mentioned in this paragraph and are shown in Table I. For example, "vib. 1D & audio" denotes the raw audio signal in combination with the euclidean norm of the raw vibration signal. Feature vectors are then min-max normalized based on values of the train set.

The DNN model consists of six fully connected layers of varying dimension. With x being the length of the input signal and n being the characteristic number of neurons, the layers are of dimensions x, n, $\frac{n}{3}$, $\frac{n}{4}$, $\frac{n}{3}$, n, x. All units use the tanh activation function and n is selected form the range 64 to 200, depending on the selected feature. For multivariant features and DNN we stack the feature vectors to achieve one-dimensional features, whereas CNN and LSTM operate directly on the two-dimensional features. Due to the high feature size (1024) compared to the number of samples, we use a rolling window of dimension 64 to create more and smaller feature vectors, that make a smaller model possible. The models threats each sub-vector independently and a decision for a original sample is created by majority vote on the sub-vectors from the rolling window.

The LSTM models consists of stacked LSTMs networks, with increasing dimension that create a two dimensional output by returning an output for every time step. The bottleneck layer reduces the dimension by returning only the last output and is followed by a repetition of the last output for every time step. Then the number of neurons is decreased opposed to the encoder. With n being the number of units, the dimensions of the layers are $n,\ \frac{n}{2},\ \frac{n}{4},\ \frac{n}{16},\ \frac{n}{16},\ \frac{n}{4}.\ \frac{n}{2},\ n$ (rounded) with a lower cap of 16 and $n = 150$.

The CNN's encoder consists of an alternating sequence of convolutional and max pooling layers, each of dimension two, with the number of filters for the layers in the encoder being $16,\ 32,\ 64,\ 128$. A fully connected layer as a bottleneck and a reversed encoder as a decoder.

We have implemented an end to end pipeline to evaluate the key requirement: "minimal configuration effort". Further, we compare our model to simple statistical benchmarks based on the reconstruction error using principal component analysis (BM PCA) (see. [9]) and an approach similar to boxplots, where values outside of the mean $\pm\ 1.5 \cdot$ iqr , with iqr being the interquartile range, are threaded as outliers (BM IQR).

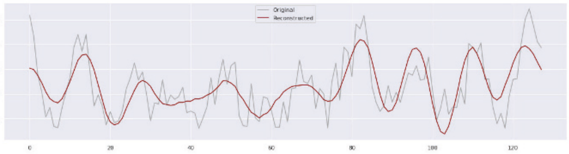


Fig. 2. Example of autoencoder input (measured signal) and output (reconstructed signal) from healthy data. The reconstruction error is derived input and output difference and used for anomaly detection.

V. Preliminary Results

We present the results of our initial experiments for all combinations of model and features in Table II and our benchmarks in Table I. In terms of F1-score, which is a trade off between precision and recall, our models beat the PCA benchmark in 15 out of 24 experiments. We achieve our best result with an LSTM network and the Euclidean norm of the 3-D vibration signal (vib. 1D) resulting in a F1-score of 0.64, a precision of 0.68 and a recall of 0.6. The PCA benchmark performed best with the 3-D vibration signal in both raw and Fourier-transformed form. Thus, our best model outperformed the benchmark by 10 %. However, the simple benchmark based on the interquartile range outperforms our autoencoders and the PCA benchmark by 1 % with a F1-Score of 0.63.

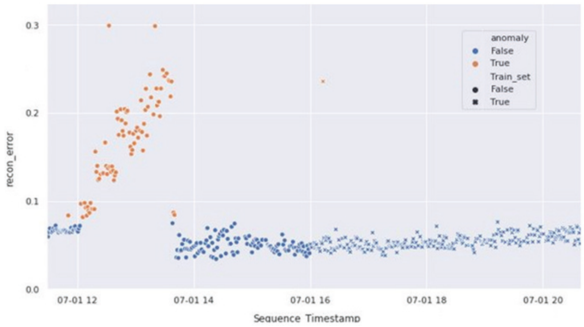


Fig. 3. The anomaly score is visualized for a sequence of measurements. A score above the threshold indicates an anomaly (orange). The shape of the points means whether we used it for training or evaluation of the autoencoder.

TABLE I
Results of benchmarks for each feature

Benchmark IQR				
Features	Acc.	F1	P	R
Vibrations 1D	0.48	**0.63**	0.47	0.94
Audio	0.56	0.31	0.55	0.21
Vibrations 3D	0.52	**0.63**	0.49	0.89
Vibrations 1D & Audio	0.47	0.62	0.46	0.94
FFT Vibrations 1D	0.48	**0.63**	0.47	0.94
FFT Audio	0.56	0.31	0.55	0.21
FFT Vibrations 3D	0.52	**0.63**	0.49	0.89
FFT Vibrations 1D & Audio	0.47	0.62	0.46	0.94
Benchmark PCA				
Features	Acc.	F1	P	R
Vibrations 1D	0.51	0.48	0.47	0.48
Audio	0.48	**0.54**	0.45	0.66
Vibrations 3D	0.50	0.45	0.46	0.44
Vibrations 1D & Audio	0.48	**0.54**	0.45	0.66
FFT Vibrations 1D	0.51	0.48	0.47	0.48
FFT Audio	0.48	**0.54**	0.45	0.66
FFT Vibrations 3D	0.50	0.45	0.46	0.44
FFT Vibrations 1D & Audio	0.48	**0.54**	0.45	0.66

VI. Discussion

Our aim was to define initial requirements for minimal-configuration anomaly detection for IIoT sensors. Based on the requirements, we focused on unsupervised machine learning and did not perform any equipment specific feature engineering. We created a hand-crafted benchmark data and made it publicity available. We experimented with three with three different neural network architectures for anomaly detection in the tube system of a peristaltic pump. Our preliminary results show an important step towards minimal-configuration anomaly detection for IIoT sensors. With all three networks we were able to outperform a benchmark based on the reconstruction error of a principal component analysis. However, it remains unclear weather the the respective sensor combination can be applied to a broad number of assets. Further, the impact of sensor position and the transferability towards identical assets remain unclear. Therefore, in the future, we will perform experiments on a data set that was recorded using different sensor positions to investigate the transferability of our models.

References

[1] D. Tolani, M. Yasar, A. Ray, and V. Yang, "Anomaly detection in aircraft gas turbine engines," *Journal of Aerospace Computing, Information and Communication*, vol. 3, no. 2, pp. 44–51, feb 2006.

[2] A. Zaher, S. McArthur, D. Infield, and Y. Patel, "Online wind turbine fault detection through automated scada data analysis," *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 12, no. 6, pp. 574–593, 2009.

TABLE II
RESULTS OF ANOMALY PREDICTION FOR EACH COMBINATION OF FEATURE SET AND MODEL.

convolutional neural network (CNN)				
Features	Acc.	F1	P	R
Vibrations 1D	0.47	0.47	0.44	0.50
Audio	0.46	0.53	0.44	0.67
Vibrations 3D	0.54	**0.62**	0.50	0.8
Vibrations 1D & Audio	0.46	0.53	0.44	0.67
FFT Vibrations 1D	0.54	0.00	0.00	0.00
FFT Audio	0.53	0.46	0.49	0.44
FFT Vibrations 3D	0.54	0.04	0.57	0.02
FFT Vibrations 1D & Audio	0.56	0.08	1.00	0.04
Fully connected neural network (DNN)				
Features	Acc.	F1	P	R
Vibrations 1D	0.49	0.53	0.46	0.61
Audio	0.46	0.54	0.45	0.68
Vibrations 3D	0.54	**0.62**	0.50	0.81
Vibrations 1D & Audio	0.47	0.54	0.45	0.68
FFT Vibrations 1D	0.59	0.47	0.59	0.39
FFT Audio	0.50	0.45	0.46	0.44
FFT Vibrations 3D	0.49	0.43	0.44	0.41
FFT Vibrations 1D & Audio	0.50	0.45	0.45	0.44
Long short-term memory neural network (LSTM)				
Features	Acc.	F1	P	R
Vibrations 1D	0.47	0.48	0.44	0.53
Audio	0.46	0.53	0.44	0.67
Vibrations 3D	0.55	**0.62**	0.51	0.79
Vibrations 1D & Audio	0.46	0.53	0.44	0.67
FFT Vibrations 1D	0.53	0.02	0.40	0.01
FFT Audio	0.52	0.48	0.48	0.48
FFT Vibrations 3D	0.60	0.37	0.71	0.25
FFT Vibrations 1D & Audio	0.63	0.42	0.74	0.30

[3] X. Tong, H. Yang, L. Wang, and Y. Miao, “The Development and Field Evaluation of an IoT System of Low-Power Vibration for Bridge Health Monitoring,” *Sensors*, vol. 19, no. 5, p. 1222, mar 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/5/1222

[4] X. Zhao, G. Wei, X. Li, Y. Qin, D. Xu, W. Tang, H. Yin, X. Wei, and L. Jia, “Self-powered triboelectric nano vibration accelerometer based wireless sensor system for railway state health monitoring,” *Nano Energy*, vol. 34, pp. 549–555, apr 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2211285517301143

[5] T. Bierweiler, H. Grieb, S. von Dosky, and M. Hartl, “Smart Sensing Environment – Use Cases and System for Plant Specific Monitoring and Optimization,” pp. 155–158, 2019. [Online]. Available: https://elibrary.vdi-verlag.de/index.php?doi=10.51202/9783181023518-155

[6] S. Su, Y. Sun, X. Gao, J. Qiu, and Z. Tian, “A correlation-change based feature selection method for iot equipment anomaly detection,” *Applied Sciences*, vol. 9, no. 3, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/3/437

[7] J. Wang, Y. Tang, S. He, C. Zhao, P. K. Sharma, O. Alfarraj, and A. Tolba, “Logevent2vec: Logevent-to-vector based anomaly detection for large-scale logs in internet of things,” *Sensors*, vol. 20, no. 9, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/9/2451

[8] Y. Kato, T. Yairi, and K. Hori, “Integrating data mining techniques and design information management for failure prevention,” in *Annual Conference of the Japanese Society for Artificial Intelligence*. Springer, 2001, pp. 475–480.

[9] J. Camacho, A. Pérez-Villegas, P. Garciá-Teodoro, and G. MacIá-Fernández, “PCA-based multivariate statistical network monitoring for anomaly detection,” *Computers and Security*, vol. 59, pp. 118–137, jun 2016.

[10] M. Sakurada and T. Yairi, “Anomaly detection using autoencoders with nonlinear dimensionality reduction,” in *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, 2014, pp. 4–11.

[11] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, “Deep autoencoding gaussian mixture model for unsupervised anomaly detection,” 2018.

[12] S. Maleki, S. Maleki, and N. R. Jennings, “Unsupervised anomaly detection with lstm autoencoders using statistical data-filtering,” *Applied Soft Computing*, vol. 108, p. 107443, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494621003665

Flexible Systems to Reach High Security Levels in the Communication with Machines and in their Maintenance

Secure Cloud Connect

Nikolaus Dürk
X-Net Services GmbH, 4020 Linz, Austria
nd@x-net.at

Abstract—**Digitalization is confronting companies and especially small and medium enterprises (SME) with an ongoing change in their environment. The use of digital technologies has a direct impact on business processes, products and services and customer behaviour. The increasing connectivity of (critical) cyber-physical objects enables the development of new applications but also leads to new safety and security related requirements in design, testing, production, maintenance and op-eration of these systems as Internet of Things (IoT) devices are always in focus of attacks.**

Production companies are currently facing enormous challenges in the area of IT security. They cannot only invest in security once, they have to consider security continuously in production en-vironments, in development of new products and during all life cycle processes. They have to define internal resources and structures as well as (external) experts to constantly work on their security strategies.

Manufacturers are forced to protect themselves from external attacks towards their production operations. At the same time they have to open their environment and closely work together with suppliers and customers. Further on their products require remote maintenance, updates and upgrades until end of life (life cycle management). Sub-suppliers of production systems and/or of components need secure access to their products in order to maintain their services and uphold quality standards. Data transfers are necessary as data source and the place of production and assembling differ.

Secure Cloud Connect (Sec3) provides highest security levels for production companies, their ma-chines and their IoT devices. Sensitive configurations and communication take place in cloud and hub systems that are owned by the companies themselves. Encrypted connection allow secure and logged remote maintenance. A quick initial installation and low time requirements during op-eration makes the solution easy applicable. Data of the machines is additionally collected, aggreg-ated and made accessible for detecting anomalies or prevention of defects. To protect sensitive data, anonymity and compliance of General Data Protection Regulation (GDPR) are ensured.

I. Motivation

We aim to address some of the major challenges and business cases for future industry. Here, our particular interests relate to Cybersecurity and safety conditions of the Internet of Things (IoT) for Cyber Physical Systems (CPS). Our focus lies on flexible tools that reach a high degree of reuse between different environments (e.g. hardware can be changed easily) while taking into account physical and energy constraints, heterogeneity of data sources and throughputs, computing power and targeted user groups. The decision for open source as basis for security was obvious as the sharing and exchange of knowledge and methods to reach customized security solutions is the most promising way into a secure future.

The increasing globalization of production and external conditions (e.g. Covid-19 restrictions, climate protection measures) complicate on-site service. The worldwide shut-downs in 2020 taught us to decentrally communicate in new dimension but also that new structures and strategies for globally distributed (production) networks are needed.

Further on, the quality of hacker attacks has increased dramatically in the recent years. Attacks on the IT infrastructure of companies are carried out by professionals with a high level of technical knowledge. Not only the Darknet has to be mentioned here as potential source of attacks, these attacks can also be carried out by competitors (e.g. to prepare an unfriendly takeover) or even by foreign state institutions.

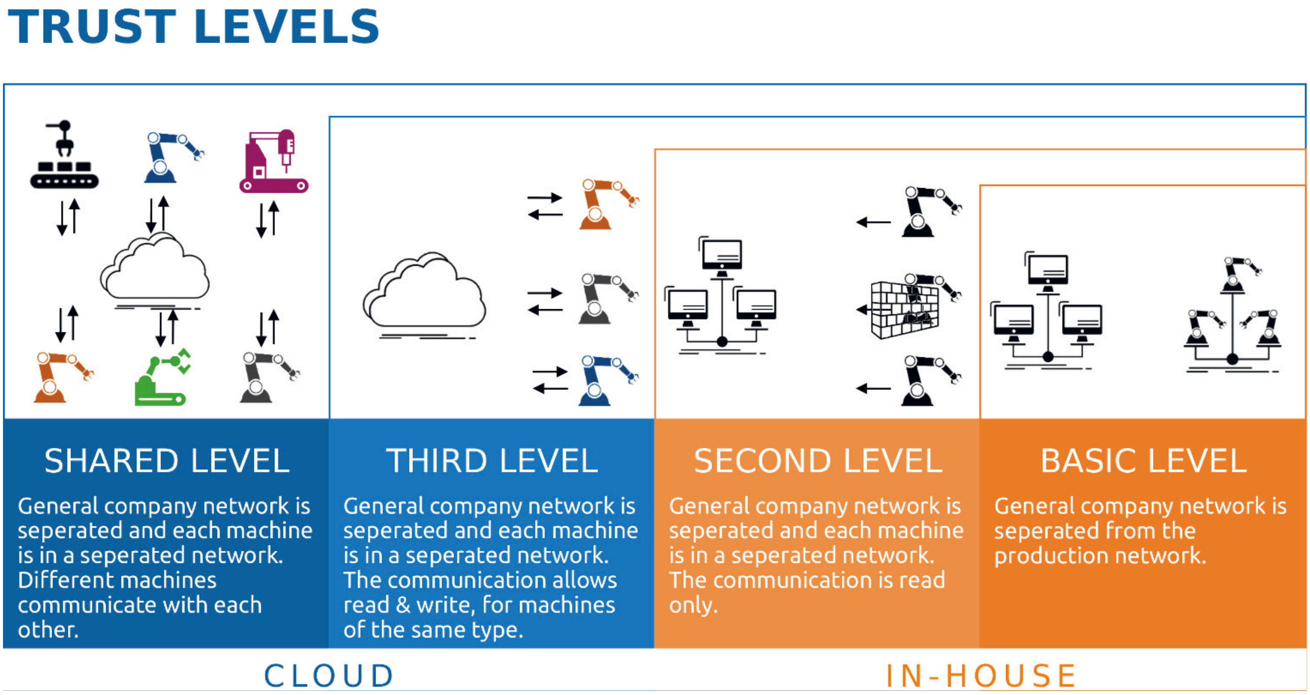
In many companies, IT systems are the backbone of business models and responsible for their performance. Problems with IT systems can cause enormous damage to a company, ranging from loss of production through loss of revenue to the existential threats. Though the expenses on IT security are still rather low until companies suffer from attacks themselves. Once they are confronted with the impact of hacking attacks, they invest significantly more in their IT security.

II. Trust Levels in production environments

Systems that isolate individual networks within the enterprise (security by isolation) use virtual networks and allow secure remote maintenance, while at the same time providing security for the machine manufacturer and the machine user through audit logs. It is important to ensure that all networks and devices are regularly maintained and receive security updates. Further on, all accesses have to be logged to get an overview about anomalies.

To reach a high level security standard, the company structure has to be divided into individual zones. Depending on the number of individual zones and the intensity of separation, different trust levels can be defined (see Fig. 1).

P. Haber et al. (Eds.), *Data Science – Analytics and Applications*,
https://doi.org/10.1007/978-3-658-36295-9_14



TRUST LEVELS

SHARED LEVEL

General company network is seperated and each machine is in a seperated network. Different machines communicate with each other.

THIRD LEVEL

General company network is seperated and each machine is in a seperated network. The communication allows read & write, for machines of the same type.

SECOND LEVEL

General company network is seperated and each machine is in a seperated network. The communication is read only.

BASIC LEVEL

General company network is seperated from the production network.

Fig. 1. Levels of trust

The **basic level** describes a simple separation between the general company network and the production network. This is the first step to enhance security, as different end devices (e.g. mobile phones, tablet PCs, notebooks), which are a fundamental weak point in a network, are separated from the production machines. Companies have to ensure that occurring incidences (e.g. single devices which are hacked or historically grown networks which have some misconfigurations) do not affect production, production areas and machine configurations.

The **second level** describes the separation between the general company network and individual networks for each machine of a production environment. User, service technician or even hacker are working only in one network and on one machine. Changes of configurations and other settings do not affect other machines or networks. The communication is read only.

Production on several production sites and locations require cloud solutions for communication. The **third level** of security describes the separation between the general company network and individual networks for each machine with additional security measures (e.g. logging of all activities, analysis of anomalies).

The highest possible security level is the **shared level**. In this level, different machines can communicate with each other, while each machine is separated in its own network. A secure cloud is available that handles the communication and all security relevant elements.

A test-bed to demonstrate how third level security can be integrated into production environments was developed during the project “IoT4CPS – Trustworthy IoT for CPS” funded by the FFG program ICT of the Future. Based on the results of the research, a reference installation could be installed at LISEC, an Austrian manufacturer of glass processing machines.

III. System Architecture

Splitting networks within an industrial environment becomes more and more important with a growing number of IoT devices. For security reasons, it is necessary to create distinct network areas that restrict the capability of IoT devices to communicate otherwise attack vectors will be opened. On the other hand, a smart production architecture must allow for easy handling of attaching, provisioning and remote maintenance of new equipment or machines, as well as data analytics for process optimization.

The concept of Security by Isolation (SBI) requires only a few hardware components, which do not necessarily have to be on-site except for a firewall system. SBI architecture provides the appropriate tools for protecting the complete communication of the individual components. It uses modern authentication and authorization methods (Active Directory, two-component authentication) as well as encryption and VPN technologies.

Sec3 factory consists of a database (core) at the mechanical engineer, several VPN hubs and the gateways (sec3 box) for the machine user. So called emergency boxes provide redundancies and take over the most important functionalities of the sec3 boxes in case of failures.

The core is the central database management system. Among other things, it manages hubs and boxes and provides informations about the connected machines, technicians, audit logs, firewall templates and rules for the boxes. Individual certificates and configurations of the boxes and the technicians who are supposed to have access to single machines are managed here. In addition, the connection of ERP, CRM or other internal company system is possible at any time.

The task of the hubs is to coordinate the operation of connections between the service technician or the machine manufacturer or the manufacturer of machine components and the machine. They are the nodes in the system and the global interfaces for the components defined in the core and establish secure and stable connections through state-of-the-art encryption technology. Sec3 requires at least two independent hubs in different locations. In case of a fail, the sec3 box is still reachable through another hub.

A connection can only be authorized via the sec3 box which takes over several functions. It is primarily used as a gateway to the connected machines, but can also act as a firewall with additional functions and tasks. The sec3 boxes are provided with IoT layers to enable IoT data collection and analysis. Another feature is a secure and fast initial installation as well as remote provisioning and maintaining. Even if the locally existing LAN or WLAN is out of service, connection is ensured by using mobile connections. The complete configuration of the sec3 boxes will be done in the core, the sec3 boxes receive the provisioning when they are switched on at the customers locations automatically. No additional configuration is necessary.

The technician is the last component in the SBI concept and the one who works in this system and carries out remote maintenance on the machines. Therefore, a certification and the allocation of permissions in the core are necessary. This ensures that a third party technician only gains access to the allowed components and not to the entire machine network. The connection of a technician to the customer network must be explicitly authorized through the customer by activating the VPN connection to the sec3 box or via a user interface. For security reasons, the access of the technician is also displayed via an USB traffic light.