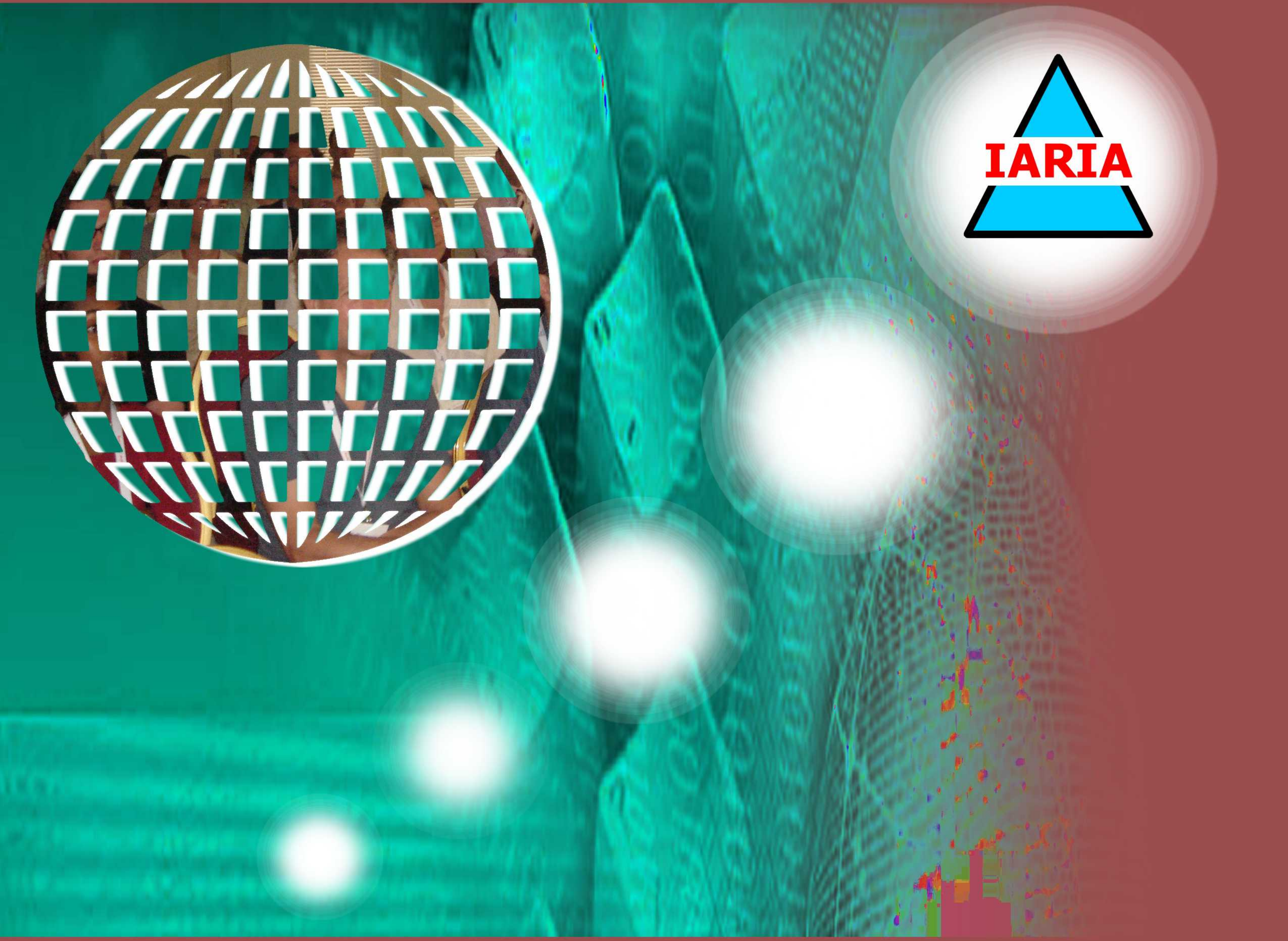
International Journal on Advances in Software



2022 vol. 15 nr. 3&4

The *International Journal on Advances in Software* is published by IARIA.
ISSN: 1942-2628
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Software
Volume 15, Number 3 & 4, 2022

Gabriele Bavota, University of Salerno, Italy
Grigorios N. Beligiannis, University of Western Greece, Greece
Noureddine Belkhatir, University of Grenoble, France
Jorge Bernardino, ISEC - Institute Polytechnic of Coimbra, Portugal
Rudolf Berrendorf, Bonn-Rhein-Sieg University of Applied Sciences - Sankt Augustin, Germany
Ateet Bhalla, Independent Consultant, India
Fernando Boronat Seguí, Universidad Politecnica de Valencia, Spain
Pierre Borne, Ecole Centrale de Lille, France
Farid Bourennani, University of Ontario Institute of Technology (UOIT), Canada
Narhimene Boustia, Saad Dahlab University - Blida, Algeria
Hongyu Pei Breivold, ABB Corporate Research, Sweden
Carsten Brockmann, Universität Potsdam, Germany
Antonio Bucchiarone, Fondazione Bruno Kessler, Italy
Georg Buchgeher, Software Competence Center Hagenberg GmbH, Austria
Dumitru Burdescu, University of Craiova, Romania
Martine Cadot, University of Nancy / LORIA, France
Isabel Candal-Vicente, Universidad Ana G. Méndez, Puerto Rico
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Jose Carlos Metrolho, Polytechnic Institute of Castelo Branco, Portugal
Alain Casali, Aix-Marseille University, France
Yaser Chaaban, Leibniz University of Hanover, Germany
Savvas A. Chatzichristofis, Democritus University of Thrace, Greece
Antonin Chazalet, Orange, France
Jiann-Liang Chen, National Dong Hwa University, China
Shiping Chen, CSIRO ICT Centre, Australia
Wen-Shiung Chen, National Chi Nan University, Taiwan
Zhe Chen, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China PR
Yoonsik Cheon, The University of Texas at El Paso, USA
Lau Cheuk Lung, INE/UFSC, Brazil
Robert Chew, Lien Centre for Social Innovation, Singapore
Andrew Connor, Auckland University of Technology, New Zealand
Rebeca Cortázar, University of Deusto, Spain
Noël Crespi, Institut Telecom, Telecom SudParis, France
Carlos E. Cuesta, Rey Juan Carlos University, Spain
Duilio Curcio, University of Calabria, Italy
Mirela Danubianu, "Stefan cel Mare" University of Suceava, Romania
Paulo Asterio de Castro Guerra, Tapijara Programação de Sistemas Ltda. - Lambari, Brazil
Cláudio de Souza Baptista, University of Campina Grande, Brazil
Maria del Pilar Angeles, Universidad Nacional Autonónoma de México, México
Rafael del Vado Vírseda, Universidad Complutense de Madrid, Spain
Giovanni Denaro, University of Milano-Bicocca, Italy
Nirmit Desai, IBM Research, India
Vincenzo Deufemia, Università di Salerno, Italy
Leandro Dias da Silva, Universidade Federal de Alagoas, Brazil
Javier Diaz, Rutgers University, USA
Nicholas John Dingle, University of Manchester, UK
Roland Dodd, CQUniversity, Australia
Aijuan Dong, Hood College, USA
Suzana Dragicevic, Simon Fraser University- Burnaby, Canada
Cédric du Mouza, CNAM, France
Ann Dunkin, Palo Alto Unified School District, USA
Jana Dvorakova, Comenius University, Slovakia

Hans-Dieter Ehrich, Technische Universität Braunschweig, Germany
Jorge Ejarque, Barcelona Supercomputing Center, Spain
Atilla Elçi, Aksaray University, Turkey
Khaled El-Fakih, American University of Sharjah, UAE
Gledson Elias, Federal University of Paraíba, Brazil
Sameh Elnikety, Microsoft Research, USA
Fausto Fasano, University of Molise, Italy
Michael Felderer, University of Innsbruck, Austria
João M. Fernandes, Universidade de Minho, Portugal
Luis Fernandez-Sanz, University of de Alcala, Spain
Felipe Ferraz, C.E.S.A.R, Brazil
Adina Magda Florea, University "Politehnica" of Bucharest, Romania
Wolfgang Fohl, Hamburg Universiy, Germany
Simon Fong, University of Macau, Macau SAR
Gianluca Franchino, Scuola Superiore Sant'Anna, Pisa, Italy
Naoki Fukuta, Shizuoka University, Japan
Martin Gaedke, Chemnitz University of Technology, Germany
Félix J. García Clemente, University of Murcia, Spain
José García-Fanjul, University of Oviedo, Spain
Felipe Garcia-Sanchez, Universidad Politecnica de Cartagena (UPCT), Spain
Michael Gebhart, Gebhart Quality Analysis (QA) 82, Germany
Tejas R. Gandhi, Virtua Health-Marlton, USA
Andrea Giachetti, Università degli Studi di Verona, Italy
Afzal Godil, National Institute of Standards and Technology, USA
Luis Gomes, Universidade Nova Lisboa, Portugal
Pascual Gonzalez, University of Castilla-La Mancha, Spain
Björn Gottfried, University of Bremen, Germany
Victor Govindaswamy, Texas A&M University, USA
Gregor Grambow, AristaFlow GmbH, Germany
Christoph Grimm, University of Kaiserslautern, Austria
Michael Grottke, University of Erlangen-Nuernberg, Germany
Vic Grout, Glyndwr University, UK
Ensar Gul, Marmara University, Turkey
Richard Gunstone, Bournemouth University, UK
Zhensheng Guo, Siemens AG, Germany
Ismail Hababeh, German Jordanian University, Jordan
Shahliza Abd Halim, Lecturer in Universiti Teknologi Malaysia, Malaysia
Herman Hartmann, University of Groningen, The Netherlands
Jameleddine Hassine, King Fahd University of Petroleum & Mineral (KFUPM), Saudi Arabia
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Peizhao Hu, NICTA, Australia
Chih-Cheng Hung, Southern Polytechnic State University, USA
Edward Hung, Hong Kong Polytechnic University, Hong Kong
Noraini Ibrahim, Universiti Teknologi Malaysia, Malaysia
Anca Daniela Ionita, University "POLITEHNICA" of Bucharest, Romania
Chris Ireland, Open University, UK
Kyoko Iwasawa, Takushoku University - Tokyo, Japan
Mehrshid Javanbakht, Azad University - Tehran, Iran
Wassim Jaziri, ISIM Sfax, Tunisia
Dayang Norhayati Abang Jawawi, Universiti Teknologi Malaysia (UTM), Malaysia
Jinyuan Jia, Tongji University. Shanghai, China
Maria Joao Ferreira, Universidade Portucalense, Portugal
Ahmed Kamel, Concordia College, Moorhead, Minnesota, USA

Teemu Kanstrén, VTT Technical Research Centre of Finland, Finland
Nittaya Kerdprasop, Suranaree University of Technology, Thailand
Ayad ali Keshlaf, Newcastle University, UK
Nhien An Le Khac, University College Dublin, Ireland
Sadegh Kharazmi, RMIT University - Melbourne, Australia
Kyoung-Sook Kim, National Institute of Information and Communications Technology, Japan
Youngjae Kim, Oak Ridge National Laboratory, USA
Cornel Klein, Siemens AG, Germany
Alexander Knapp, University of Augsburg, Germany
Radek Koci, Brno University of Technology, Czech Republic
Christian Kop, University of Klagenfurt, Austria
Michal Krátký, VŠB - Technical University of Ostrava, Czech Republic
Narayanan Kulathuramaiyer, Universiti Malaysia Sarawak, Malaysia
Satoshi Kurihara, Osaka University, Japan
Eugenijus Kurilovas, Vilnius University, Lithuania
Alla Lake, Linfo Systems, LLC, USA
Fritz Laux, Reutlingen University, Germany
Luigi Lavazza, Università dell'Insubria, Italy
Fábio Luiz Leite Júnior, Universidade Estadual da Paraiba,Brazil
Alain Lelu, University of Franche-Comté / LORIA, France
Cynthia Y. Lester, Georgia Perimeter College, USA
Clement Leung, Hong Kong Baptist University, Hong Kong
Weidong Li, University of Connecticut, USA
Corrado Loglisci, University of Bari, Italy
Francesco Longo, University of Calabria, Italy
Sérgio F. Lopes, University of Minho, Portugal
Pericles Loucopoulos, Loughborough University, UK
Alen Lovrencic, University of Zagreb, Croatia
Qifeng Lu, MacroSys, LLC, USA
Xun Luo, Qualcomm Inc., USA
Stephane Maag, Telecom SudParis, France
Ricardo J. Machado, University of Minho, Portugal
Maryam Tayefeh Mahmoudi, Research Institute for ICT, Iran
Nicos Malevris, Athens University of Economics and Business, Greece
Herwig Mannaert, University of Antwerp, Belgium
José Manuel Molina López, Universidad Carlos III de Madrid, Spain
Francesco Marcelloni, University of Pisa, Italy
Eda Marchetti, Consiglio Nazionale delle Ricerche (CNR), Italy
Gerasimos Marketos, University of Piraeus, Greece
Abel Marrero, Bombardier Transportation, Germany
Adriana Martin, Universidad Nacional de la Patagonia Austral / Universidad Nacional del Comahue, Argentina
Goran Martinovic, J.J. Strossmayer University of Osijek, Croatia
Paulo Martins, University of Trás-os-Montes e Alto Douro (UTAD), Portugal
Stephan Mäs, Technical University of Dresden, Germany
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Jose Merseguer, Universidad de Zaragoza, Spain
Seyedeh Leili Mirtaheri, Iran University of Science & Technology, Iran
Lars Moench, University of Hagen, Germany
Yasuhiko Morimoto, Hiroshima University, Japan
Antonio Navarro Martín, Universidad Complutense de Madrid, Spain
Filippo Neri, University of Naples, Italy
Muaz A. Niazi, Bahria University, Islamabad, Pakistan
Natalja Nikitina, KTH Royal Institute of Technology, Sweden

George Spanoudakis, City University London, UK
Cristian Stanciu, University Politehnica of Bucharest, Romania
Lena Strömbäck, SMHI, Sweden
Osamu Takaki, Japan Advanced Institute of Science and Technology, Japan
Antonio J. Tallón-Ballesteros, University of Seville, Spain
Wasif Tanveer, University of Engineering & Technology - Lahore, Pakistan
Ergin Tari, Istanbul Technical University, Turkey
Steffen Thiel, Furtwangen University of Applied Sciences, Germany
Jean-Claude Thill, Univ. of North Carolina at Charlotte, USA
Pierre Tiako, Langston University, USA
Božo Tomas, HT Mostar, Bosnia and Herzegovina
Davide Tosi, Università degli Studi dell'Insubria, Italy
Guglielmo Trentin, National Research Council, Italy
Dragos Truscan, Åbo Akademi University, Finland
Chrisa Tsinaraki, Technical University of Crete, Greece
Roland Ukor, FirstLinq Limited, UK
Torsten Ullrich, Fraunhofer Austria Research GmbH, Austria
José Valente de Oliveira, Universidade do Algarve, Portugal
Dieter Van Nuffel, University of Antwerp, Belgium
Shirshu Varma, Indian Institute of Information Technology, Allahabad, India
Konstantina Vassilopoulou, Harokopio University of Athens, Greece
Miroslav Velev, Aries Design Automation, USA
Tanja E. J. Vos, Universidad Politécnica de Valencia, Spain
Krzysztof Walczak, Poznan University of Economics, Poland
Yandong Wang, Wuhan University, China
Rainer Weinreich, Johannes Kepler University Linz, Austria
Stefan Wesarg, Fraunhofer IGD, Germany
Wojciech Wiza, Poznan University of Economics, Poland
Martin Wojtczyk, Technische Universität München, Germany
Hao Wu, School of Information Science and Engineering, Yunnan University, China
Mudasser F. Wyne, National University, USA
Zhengchuan Xu, Fudan University, P.R.China
Yiping Yao, National University of Defense Technology, Changsha, Hunan, China
Stoyan Yordanov Garbatov, Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento, INESC-ID, Portugal
Weihai Yu, University of Tromsø, Norway
Wenbing Zhao, Cleveland State University, USA
Hong Zhu, Oxford Brookes University, UK
Martin Zinner, Technische Universität Dresden, Germany

International Journal on Advances in Software

Volume 15, Numbers 3 & 4, 2022

CONTENTS

pages: 141 - 151
Improving Image Tracing with Convolutional Autoencoders by High-Pass Filter Preprocessing
Zineddine Bettouche, Deggendorf Institute of Technology, Germany
Andreas Fischer, Deggendorf Institute of Technology, Germany

pages: 152 - 164
An In-depth Comparison of Experiment Tracking Tools for Machine Learning Applications
Tim Budras, Karlsruhe University of Applied Sciences, Germany
Maximilian Blanck, inovex GmbH, Germany
Tilman Berger, inovex GmbH, Germany
Andreas Schmidt, Karlsruhe University of Applied Sciences & Karlsruhe Institute of Technology, Germany

pages: 165 - 174
Automatic Generation of Geographically Accurate Bus Route Maps and its Evaluation
Daisuke Yamamoto, Nagoya Institute of Technology, Japan
Yonghwan Kim, Nagoya Institute of Technology, Japan
Sogo Mizutani, Nagoya Institute of Technology, Japan

pages: 175 - 187
State of Affair in Terms of Big Data Utilization in Complex System Engineering Organizations: A Case Study in the Context of Norwegian Industry
Fahim Salim, University of South-Eastern Norway, Norway
Haytham Ali, University of South-Eastern Norway, Norway
Tommy Langen, University of South-Eastern Norway, Norway
Pragna Deb, University of South-Eastern Norway, Norway
Andreas Wettre, Oslo School of Architecture and Design, Norway
Gerrit Muller, University of South-Eastern Norway, Norway
Kristin Falk, University of South-Eastern Norway, Norway

pages: 188 - 199
Optimizing Remediation of Spatially Dispersed Contaminated Parcels under an Annual Budget Constraint
Floris Abrams, Belgian Nuclear Research Centre (SCK CEN) / Katholieke Universiteit Leuven (KU Leuven), Belgium
Lieve Sweeck, Belgian Nuclear Research Centre (SCK CEN), Belgium
Johan Camps, Belgian Nuclear Research Centre (SCK CEN), Belgium
Dirk Cattrysse, Katholieke Universiteit Leuven (KU Leuven), Belgium
Jos Van Orshoven, Katholieke Universiteit Leuven (KU Leuven), Belgium

pages: 200 - 210
Green Storage: Parallel File Systems on ARM
Timm Leon Erxleben, Otto von Guericke University Magdeburg, Germany
Kira Duwe, Otto von Guericke University Magdeburg, Germany
Jens Saak, Max Planck Institute for Dynamics of Complex Technical Systems, Germany
Martin Köhler, Max Planck Institute for Dynamics of Complex Technical Systems, Germany
Michael Kuhn, Otto von Guericke University Magdeburg, Germany

pages: 211 - 223
Using Locally Weighted Regression to Estimate the Functional Size of Software: an Empirical Study
Luigi Lavazza, Università degli Studi dell'Insubria, Italy
Angela Locoro, Università degli Studi dell'Insubria, Italy
Geng Liu, Hangzhou Dianzi University, China
Roberto Meli, DPO, Italy

pages: 224 - 234
Database Technology Evolution
Malcolm Crowe, University of the West of Scotland, United Kingdom
Fritz Laux, Universität Reutlingen, Germany

pages: 235 - 249
Designing Context-aware Data Plausibility Automation Using Machine Learning
Mohaddeseh Basiri, KTH Royal Institute of Technology, Sweden
Johannes Himmelbauer, Software Competence Center Hagenberg GmbH, Austria
Lisa Ehrlinger, Software Competence Center Hagenberg GmbH, Austria
Mihhail Matskin, KTH Royal Institute of Technology, Sweden

Improving Image Tracing with Convolutional Autoencoders by High-Pass Filter Preprocessing

Zineddine Bettouche and Andreas Fischer
Deggendorf Institute of Technology
Dieter-Görlitz-Platz 1
94469 Deggendorf
E-Mail: zineddine.bettouche@th-deg.de, andreas.fischer@th-deg.de

***Abstract*—The process of transforming a raster image into a vector representation is known as image tracing. This study looks into several processing methods that include high-pass filtering, autoencoding, and vectorization to extract an abstract representation of an image. According to the findings, rebuilding an image with autoencoders, high-pass filtering it, and then vectorizing it can represent the image more abstractly while increasing the effectiveness of the vectorization process.**

***Index Terms*—image quality; vector graphics; principal component analysis; neural networks; autoencoders; high-pass filters; vectorization; complexity theory; and information technology.**

I. Introduction

Object recognition is considered a complex task in the processing field. Its complexity far exceeds simple arithmetic operations. With the massive amount of data generated each year, manual calculations done by hand are completely ignored. Therefore, data processing and evaluation are automated for all operations.

In recent years, many studies have emerged to contribute to the advancement of knowledge in the field of object recognition. Two of the pillars of this field are image processing and artificial intelligence (AI). AI is a fascinating subject that has attracted a lot of attention in the last decade, especially with its use in computer vision. Now not only filter-based models, e.g., Haar Cascade, can be trained to classify images, but also neural networks can be wired to learn how to detect various shapes and objects. The models generally learn from the pixel values and model their structures in mathematical equations, which begs the question of whether it would be more efficient for the models to learn from vector images as they are closer to the nature of the trained models than spatial data in the form of pixel arrays. Thus, this article is an attempt to improve the tracing of images by using autoencoders and high-pass filters to obtain an abstract representation of images in vector form. The highpass filters are chosen since they emphasize the important features of an image. This work is considered a step forward to achieving a better training rate in object recognition with ANN.

This paper is an extended version of the previously published paper [1]that discussed the summarized content of the findings that this work produced. A more in-depth discussion about the techniques used in the work, such as Image Tracing, Potrace, and autoencoders, has been added in the background section. The previous papers that touched on the topic of image tracing have been further discussed in detail to illustrate the place our work takes in relation to what has already been accomplished in the field. Concerning the methodology followed, a detailed description of the autoencoding network built is provided, and the choice of the layers is justified. When it comes to the experimentation part, other experiments are added, such as the attempt at reducing the noise without blurring the images. The experiments already introduced in the previous paper are extended to further discuss their findings, and detailed images that visualize those findings are added.

In other words, concerning the added value of this paper over its previous conference version, it can be stated that every section has gone through many further details, to present a richer methodology section (as for the ease of future building over our findings), to underline the networks built, and technologies used (such as the trained autoencoders that were described layer by layer and Potrace as a vectorization tool), and to provide an extended experimentation section, as the experiments' discussions are lengthened, detailed more with visualization of their results, and assisted with other experiments (such as a blur-free noise reduction attempt).

At first, there was the question: if the autoencoding of an image can improve its vectorized format by reconstructing its important features, how can high-pass filters come into play in the process? In other words, ”Can a high-pass filter be used in combination with an autoencoding model to achieve an abstract representation of the image through the process of vectorization?” Thus, various ideas branched from this node, leading to the different pipelines that can be built to experiment with high-pass filter integration. For instance, the filters can be put before the autoencoding stage of a model that is already trained with filtered images to better reconstruct the significant data, leading to better vectorization. More

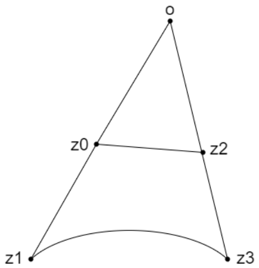


Figure 1. Bezier curve

systematically, the autoencoding stage can act as a smoothing process, removing the noise from the images while reducing their complexity, while the filters come afterward to further enhance the quality of the important features, leading to a more abstract representation.

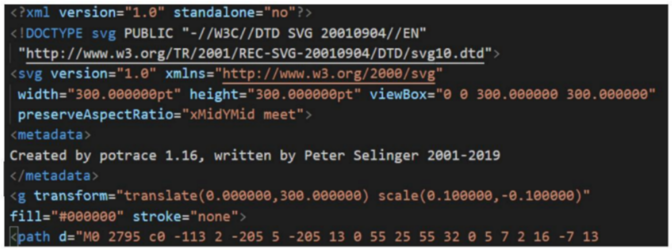
The remainder of this paper is structured as follows: In Section II, an introduction is given to image tracing, autoencoders, and high-pass filters. Section III discusses related work. Section IV introduces the methodology of this paper, including the evaluation methods used and the reasons why they were chosen. Section V presents the experiments and their results. This is the part that attempts to eliminate inefficient processing algorithms so that only a few pipelines that score closely are put forward for further evaluation. Section VI includes the evaluation of the different processing pipelines built and closes with a summarizing interpretation. Finally, Section VII concludes the paper and discusses future work.

II. Background

A. Image Tracing

Image tracing is the process of converting a bitmap into a vector graphic. As Selinger writes in his tracing algorithm [2], vector graphics are described as algebraic formulas of the contours, typically in the form of Bezier curves. The advantage of displaying an image as a vector outline is that it can be scaled to any size without loss of quality. They are independent of the resolution and are used, for example, for fonts, since these must be available in many different sizes. However, most input and output devices, such as scanners, displays, and printers, generate bitmaps or raster data. For this reason, a conversion between the two formats is necessary. Converting a vector graphic into a bitmap is called ”rendering” or ”rasterizing.” Tracing algorithms are inherently imperfect because many possible vector outlines can represent the same bitmap. Of the many possible vector representations that can result in a particular bitmap, some are more plausible or aesthetically pleasing than others. For example, to render bitmaps with a high resolution, each black pixel is represented as a precise square that creates staircase patterns. However, spikes are neither pleasant to look at nor are they particularly plausible interpretations of the original image. Bezier curves are used to represent the outlines.

As seen in Figure 1, a cubic Bezier curve consists of four control points, which determine the curvature of the curve. As a rule, the vector graphics are saved as SVG files (Scalable Vector Graphics). This file format is a special form of an XML file. XML stands for Extensible Markup Language. It is used to present hierarchically structured data in a human-readable format. As can be seen from Figure 2, the structure of this file is based on the Extensible Markup Language scheme. The file header defines which versions of XML and SVG are used. The height and width of the graphic in points are also specified. In this case, the g element represents the drawing area on which to draw. The elements to be drawn consist of tags stored as XML elements. They are particularly important in connection with the path elements. Quadratic and cubic Bezier curves, as well as elliptical arcs and lines, can be put together as best fits. The entries here determine which form the path takes.



```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20010904//EN"
 "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
<svg version="1.0" xmlns="http://www.w3.org/2000/svg"
 width="300.000000pt" height="300.000000pt" viewBox="0 0 300.000000 300.000000"
 preserveAspectRatio="xMidYMid meet">
<metadata>
Created by potrace 1.16, written by Peter Selinger 2001-2019
</metadata>
<g transform="translate(0.000000,300.000000) scale(0.100000,-0.100000)"
fill="#000000" stroke="none">
<path d="M0 2795 c0 -113 2 -205 5 -205 13 0 55 25 55 32 0 5 7 2 16 -7 13
```

Figure 2. Header of an SVG file by potrace

B. Potrace as a Vectorization Tool

Potrace is a tracing algorithm that was developed by Peter Selinger [2]. It is considered simple and efficient as it produces excellent results. Potrace stands for ”polygon tracer,” where the output of the algorithm is not a polygon but a contour made of Bezier curves. This algorithm works particularly well for high-resolution images. Potrace generates grayscale images as a threshold vector rather than as the output. The conversion from a bitmap to a vector graphic is done in several steps. First, the bitmap is broken down into several paths that form the boundaries between black and white areas. The points adjoining four pixels are given integer coordinates. These points are saved as vertices when the four adjacent pixels are not the same color. The connection between two vertices is called the edge. A path is thus a sequence of vertices, whereby the edges must all be different. The path composition in Potrace works by moving along the edges between the pixels. Every time a corner is found, a decision is made as to which direction the path will continue based on the colors of the surrounding pixels. If a closed path is defined, it is removed from the image by inverting all pixel colors inside the path. This will define a new bitmap on which the algorithm will be applied recursively until there are no more black pixels. Then its optimal polygon is approximately determined for each path. The criterion for optimality with Potrace is the number of segments. A polygon with a few segments is therefore more optimal than one with several segments. In the last phase, the polygons obtained are converted into a smooth vector outline. Here, the vertices are first corrected so that they correspond as closely as possible to the original bitmap. Furthermore, in the

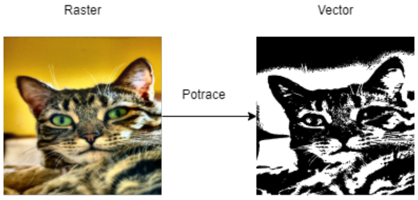


Figure 3. Potrace vectorization

main step, the corners and curves are calculated based on the length of the adjacent segments and the angles between them. Optionally, the curves can be optimized after this process so that they match the original bitmap as closely as possible. Then, in the main step, the corners and curves are calculated based on the length of the adjacent segments and the angles between them. Optionally, the curves can be optimized after this process so that they match the original bitmap even more closely. Then, in the main step, the corners and curves are calculated based on the length of the adjacent segments and the angles between them. Finally, the curves can be optimized after this process. Figure 3 shows the output vector image when applying Potrace to an input raster image.

C. Autoencoder

A typical use of a neural network is for supervised learning. It involves training data, which contains an output label. The neural network tries to learn the mapping from the given input to the given output label. Nevertheless, if the input vector itself replaces the output label, then the network will try to find the mapping from the input to itself. This would be the identity function, which is a trivial mapping. However, if the network is not allowed to simply copy the input, then it will be forced to capture only the salient features. This constraint opens up a different field of applications for neural networks, which was unknown. The primary applications are dimensionality reduction and specific data compression. The network is first trained on the given input. The network attempts to reconstruct the given input from the features it has picked up and outputs an approximation of the input. The training step involves the computation of the error and backpropagating the error. The typical architecture of an autoencoder resembles a bottleneck. Figure 4 depicts the schematic structure of an autoencoder.

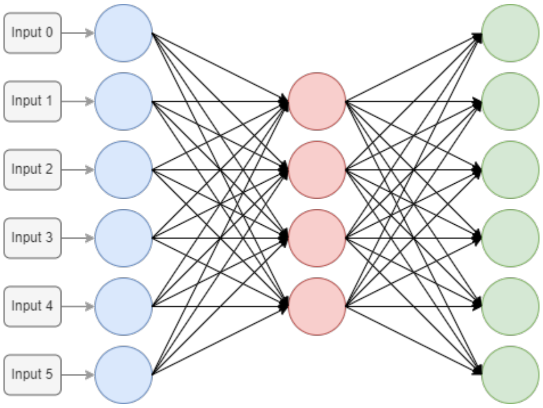


Figure 4. Example structure of an autoencoding network

The encoder part of the network is used for encoding and sometimes even for data compression purposes, although it is not very effective as compared to other general compression techniques like JPEG. Encoding is achieved by the encoder part of the network, which has a decreasing number of hidden units in each layer. Thus, this part is forced to pick up only the most significant and representative features of the data. The second half of the network performs the decoding function. This part has an increasing number of hidden units in each layer and thus tries to reconstruct the original input from the encoded data. Therefore, autoencoders are an unsupervised learning technique. Training an autoencoder for data compression: For a data compression procedure, the most important aspect of compression is the reliability of the reconstruction of the compressed data. This requirement dictates the structure of the autoencoder as a bottleneck.

1) **Encoding the input data:** The autoencoder first tries to encode the data using the initialized weights and biases.
2) **Decoding the input data:** The autoencoder tries to reconstruct the original input from the encoded data to test the reliability of the encoding.
3) **Backpropagating the error:** After the reconstruction, the loss function is computed to determine the reliability of the encoding. The error generated is backpropagated. The above-described training process is reiterated several times until an acceptable level of reconstruction is reached.

After the training process, only the encoder part of the autoencoder is retained to encode a similar type of data used in the training process. The different ways to constrain the network are:

- **Keep small Hidden Layers:** If the size of each hidden layer is kept as small as possible, then the network will be forced to pick up only the representative features of the data thus encoding the data.
- **Regularization:** In this method, a loss term is added to the cost function which encourages the network to train in ways other than copying the input.
- **Denoising:** Another way of constraining the network is to add noise to the input and teach the network how to remove the noise from the data.
- **Tuning the Activation Functions:** This method involves changing the activation functions of various nodes so that a majority of the nodes are dormant thus effectively reducing the size of the hidden layers.

D. High-pass Filters

A high-pass filter can be used to make an image appear sharper. These filters (e.g., Sobel [3] and Canny [4]) emphasize fine details in the image. The change in intensity is used by high-pass filtering. If one pixel is brighter than its immediate neighbors, it gets boosted. Figure 5 shows the result of applying a high-pass filter (Sobel) on a random image.

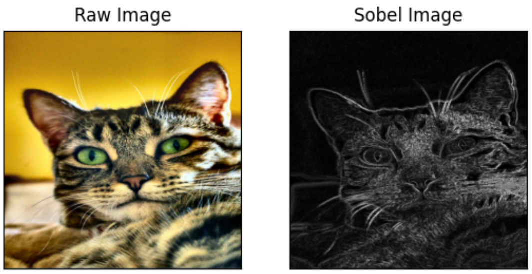


Figure 5. Applying Sobel derivatives on a random image

III. Related Work

Image segmentation can be considered an extension of image classification where localization succeeds the classification process. It is a superset of image classification with the model pinpointing where a corresponding object is present by outlining the object's boundary. Image segmentation techniques can be divided into two classes:

- Classical computer vision approaches: such as thresholding, edge, region- or cluster-based segmentation techniques.
- AI-based approaches using mainly autoencoders. For instance, DeepLab made use of convolutions to replace simple pooling operations and prevent significant information loss while downsampling.

In our paper, we focus on the use of high-pass filters with autoencoders, which succeeded with a vectorization process. Hence, the relevant work on these topics is introduced in this section.

To create better vectorize vectors, Lu et al. [5] leverage additional depth information stored in RGB-D images. Although they anticipate consumer gear will soon be able to produce photos with depth information, this still has to happen. The method described here, however, operates with standard RGB photos without the need for additional gear.

Bera [6] offers a different method for image vectorization. It emphasizes the advancement made possible by edge detection techniques. This study, in contrast, looks into the benefits of dimensionality reduction.

A method for vector pictures based on splines rather than Bézier curves is presented by Chen et al. [7] To create a combination of raster and vector graphics, they concentrate on data structures that facilitate real-time editing.

Solomon and Bessmeltsev [8] investigated the usage of frame fields in an MIT study. Finding a smooth frame field on the image plane with at least one direction aligned with neighboring drawing outlines is the basic goal of their method. The two directions of the field will line up with the two intersecting contours at X- or T-shaped junctions. The frame field is then traced, and the traced curves are then grouped into strokes to extract the drawing's topology. Finally, they produced a vectorization that was in line with the frame field using the extracted topology.

Lacroix [9] examined several R2V conversion issues, and a method utilizing a preprocessing stage that creates a mask from which edges are eliminated and lines are retained has been suggested. Then clustering is carried out using only the pixels from the mask. In this situation, a novel algorithm called the median shift has been suggested. The labeling procedure that follows should also take into account the type of pixel. The final stage entails a regularization process. In various examples, the significance of the pre-processing ignoring edge pixels while keeping lines has been demonstrated. Additionally, tests demonstrated the superiority of the median shift over both the mean shift and the Vector-Magic clustering method. This paper also showed that better line vectorization can be obtained by enabling the extraction of dark lines, which can support the use of high-pass filters as a preprocessing stage to put further emphasis on those dark lines.

On the straightforward job of denoising additive white Gaussian noise, Xie et al. [10] developed a unique strategy that performs on par with conventional linear sparse coding algorithms. In the process of fixing damaged photos, autoencoders are used to lower image noise.

An approach that completes the automatic extraction and vectorization of the road network was presented by Gong et al. [11], first, varied sizes and strong connection; second, complicated backgrounds and occlusions; and third, high resolution and a limited share of roads in the image are the key barriers to extracting roads from remote sensing photos. Road network extraction and vectorization preservation make up the two primary parts of the road vectorization technique in this paper. This study also demonstrates the benefits of dense dilation convolution, indicating the potential for adopting autoencoding models to maintain vectorization.

Fischer and Amesberger [12] showed that preprocessing the raster image with an autoencoder neural network can reduce complexity by over 70% while keeping a reasonable image quality. They proved that autoencoders perform significantly better compared to PCA in this task. We base our work on this previous work, having a closer look at the effect of high-pass filters on autoencoding in an image vectorization pipeline.

IV. Methodology

In this section, the general approach is described. First, the selected dataset is introduced. The structure of the employed autoencoder is explained next. Details about the software implementation are given, and the processing pipeline is highlighted. Finally, evaluation methods are discussed.

A. CAT Dataset - as Data

A dataset with over 10,000 cat images is used as the basis for training the autoencoder for evaluating the results. The CAT dataset was published in 2008 by Zhang et al. [13]. The content of the images is secondary for this work: The main reason this dataset is used is the fact that features such as ears, eyes, and noses are relatively easy to see in these images. The autoencoding model can thus be trained on these features and reliably reproduce them.

B. Autoencoder - Functional Structure

The starting point is input with the size 256 x 256 x 1 (a 256 x 256 grayscale image). The first layer of the autoencoder is a convolution layer that contains 16 different trainable filter kernels. Each kernel can result in a different representation of the input image. A Max-Pooling layer is connected to the convolutional layer to increase the density of the data and reduce the necessary computing power by reducing the number of trainable neurons. This 2x2 layer halved the size of the original image. This convolutional-max-pooling layer cascade is repeated twice for the next two layers, with the convolutional layer having 8 different filters and the same 2x2 max-pooling layer resulting in 64x64 and 32x32 sizes. In the last convolution layer of the encoder, which receives a 32x32 matrix as input, only four convolution kernels are used. The point of highest data density is here reached; therefore, the Max-Pooling layer is omitted. This layer of the autoencoder contains the most compact coding or representation of the data set. Figure 6 shows the encoder part of the autoencoder.

The decoder follows the layer with the highest data density. This part of the autoencoder is responsible for reconstructing the learned encoding. It uses transposed convolution layers and batch normalization layers. The transposed convolution layer works in a similar way to a convolution layer. The difference between the two is that by transposing the input, the layer is no longer compressed but decompressed. Here, the principles of the convolution layer are reversed. The filter kernel is used to determine how the input value is broken down into the larger grid. By using this layer, the image matrix is again enlarged. The transposed convolution layer is followed by a batch normalization layer. These layers, also known as batch norms, serve to accelerate and stabilize the learning process of neural networks. They reduce the amount by which the values of the neurons can shift. On the one hand, the network can train faster because the batch norm ensures that the activation value is neither too high nor too low. On the other hand, using this layer also reduces overfitting since less information is lost through dropouts.

The decoder connects directly to the encoder to take over the most compact representation of the data set passed by the encoding layers. First, the decoder receives a tensor with a size of 32x32x4 as input. The first function that is applied to this tensor is a transposed convolution layer. This results in an enlargement of the image matrix to 64x64. Four 3x3 filter kernels are used here. This is followed by a batch-norm layer to normalize the results and accelerate the learning process. The same process is repeated with a different number of filter kernels to maintain the symmetrical structure of the autoencoder after reaching the original matrix size of 256x256; another transposed convolution layer is added. This ensures that the output of the first layer and the input of the last layer have the same size. The final layer reduces the tensor dimension to one to produce a grayscale image as output. Figure 7 shows the decoder part of the autoencoder.

C. Software Implementation

The test/evaluation framework was implemented in Python. The autoencoder was implemented with TensorFlow [14] and Keras [15]. The convolutional neural network was built with convolution and pooling layers in three steps to a 32×32 bottleneck. The decoder mirrors this structure with three steps of transposed convolutional layers and batch normalization layers. The autoencoder input is set to a 255x255 image (gray-scaled). The high-pass filters used in this paper are the standard implementations in OpenCV [16].

D. General Approach of Processing

Regardless of the path an image takes in any pipeline that will be built, the first processing stage is always going to be converting the image into grayscale. The focus of this work is on single-channel images; however, it can be extended in the future for multi-channel (RGB) processing. Therefore, when a pipeline is demonstrated visually, the initial version of the image displayed is going to be grayscale, but this is implying that the raw RGB images were all grayscaled, which will be a common branch for all the pipelines built in this work.

After an image is grayscaled, it will go through a certain cascade of processing stages. In this paper, the stages concerned are high-pass filtering, autoencoding, and vectorization. The experiments in this work are going to tune the different parameters that these stages can take. More importantly, the outputs of all pipelines possible are going to be in a vector format because we are attempting to enhance the vectorization process while aiming for an abstract representation of the image. Therefore, a rasterization stage is going to always be placed at the end of every pipeline. Converting images back into their raster format is mandatory to perform a comparison between the grayscale image that was initially fed to a pipeline and its resulting vector format. Hence, we rasterize the vector output to be able to evaluate the efficiency of the pipeline. A general processing approach for the different pipelines is shown in Figure 8.

E. Evaluation Methods

The case at hand deals with both vector and raster images. Therefore, for a comparison to take place, a comparison method for each format needs to be selected.

- **Vector:** Various methods can be used to measure the level of complexity in a vector image. One is the file size, which can be used to calculate the length of all path entries in the file. Furthermore, investigating the reduction of complexity can be done by analyzing the longest path tags. The number of path tags can be taken as a characteristic value of the complexity. In this paper, it is assumed that the number of SVG path entries is directly related to its complexity.
- **Raster:** There are mainly two common ways of comparing raster images. The first one is comparing images based on the mean squared error (MSE) [17]. The MSE value denotes the average difference of the pixels all over the image. A higher MSE value designates a greater

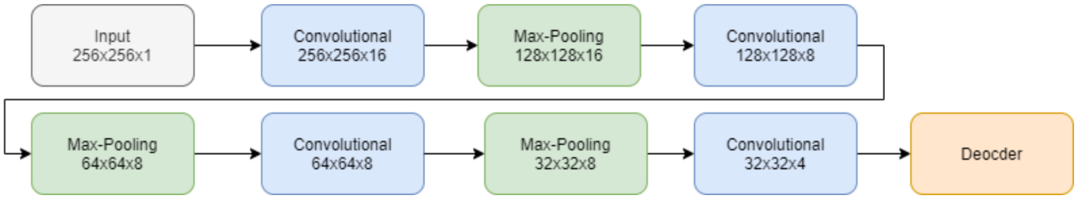


Figure 6. Encoder part of autoencoder

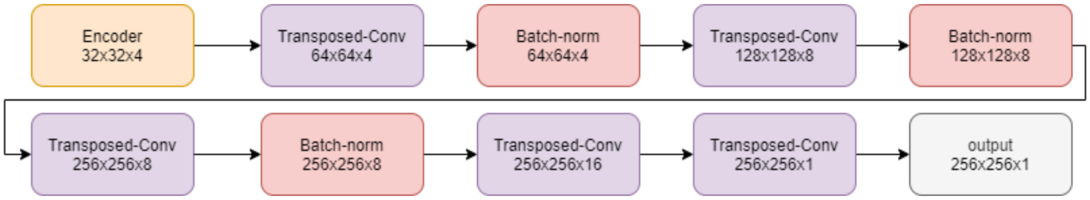


Figure 7. Decoder part of autoencoder

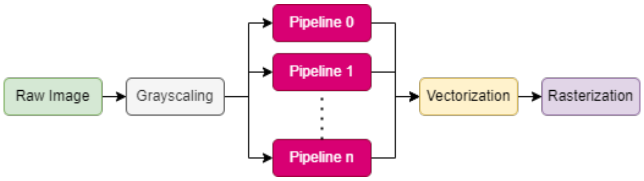


Figure 8. General processing approach

difference between the original image and the processed image. Nonetheless, it is indispensable to be extremely careful with the edges. A major problem with the MSE is that large differences between the pixel values do not necessarily mean large differences in the content of the images. The Structural Similarity Index (SSIM) [18] is used to account for changes in the structure of the image rather than just the perceived change in pixel values across the entire image. The implementation of the SSIM used is contained in the Python library Scikit-image (also known as ”Scikit”) [19]. The SSIM method is significantly more complex and computationally intensive than the MSE method, but essentially, the SSIM tries to model the perceived change in the structural information of the image, while the MSE estimates the perceived errors.

In the experiments conducted for this paper, the results of MSE and SSIM drive the same conclusion. Therefore, to avoid redundancy, only the SSIM graphs are displayed in this paper.

V. Experimentation

Firstly, a sample of five images was filtered with the initial high-pass filters. The results are shown in Figure 9.

The first impression is that the Gaussian filter results in some significant noise. Both the Sobel and the Canny filters were acceptable, with the Sobel seemingly having better results for the human eye. Because it made more sense to have the detected lines drawn black on a white image than the opposite case, the three filters were inverted.

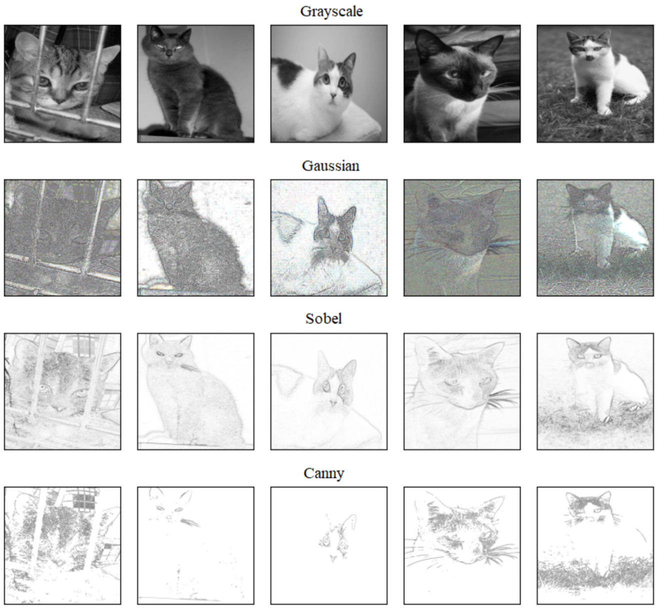


Figure 9. Applying different filters to five random images

A. Blur-Free Noise-Reduction Filtering

In an attempt to reduce the noise the Gaussian filter was causing, two trials were done. They both worked by cascading a filter on top of each high-pass filter. This smoothing filter should result in noise reduction while avoiding blurring the image. Hence, two filters were chosen: difference and grain-extract filters. Figure 10 shows the result of applying the two chosen filters on the high-pass filters.

Although the image is still too noisy to be fed into a neural network, the noise-reduction filters may provide a roughly improved version of the Gaussian filter. The difference and

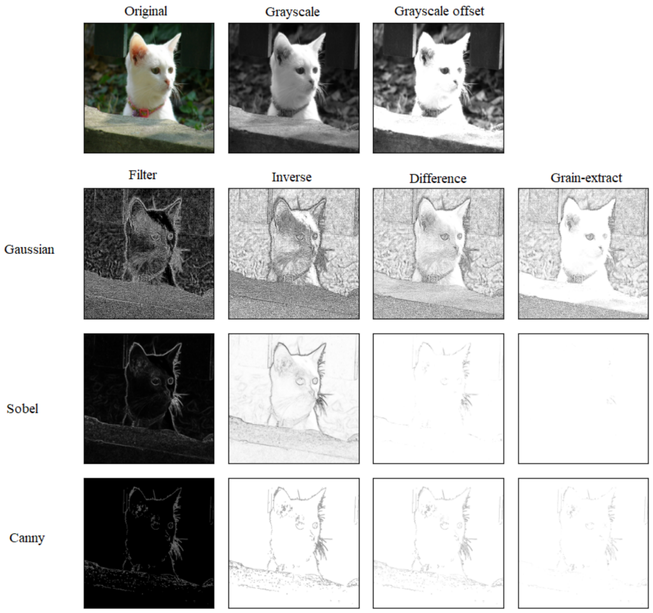


Figure 10. Applying the difference and grain-extract to a random image after being filtered

grain-extract filters, however, resulted in a decline in image quality and a sizable data loss as compared to the Sobel and Canny filters. The experiment therefore suggests that these two recommended filters are unsuitable for use in a subsequent preprocessing stage and that the Gaussian filter should be categorically excluded from any further use in the project due to its inherent noise.

B. Filter-Inversion Effect on Autoencoding

The second experiment done in this section is obtaining the difference between training an autoencoder with images whose lines are drawn in black on a white background and training it with the same images but inverted.

Therefore, four models of autoencoders were trained with 5000 epochs each in addition to the default model, which makes them five models each trained with the following types of images respectively: grayscale images, Sobel-direct images, Sobel-inverse images, canny-direct images, and canny-inverse images (direct: dark background and white features. inverse: inverse of direct). Five images were selected randomly and put through the five trained models as shown in Figure 11.

The first conclusion drawn was that, when training an autoencoder, the semi-supervised neural network responds better when the training images have darker lines in their important features. However, a rough estimation with the human eye would not do, but rather an exact mathematical calculation. Therefore, a measurement of similarity was done between every image and its decoded version. This was a better way of using the SSIM than comparing them with the default images, as the goal was to determine how close the autoencoding was. For this part, 50 images were used to dampen the image-specific features and make the measurement more generalized.The measured values were plotted in Figure 12.

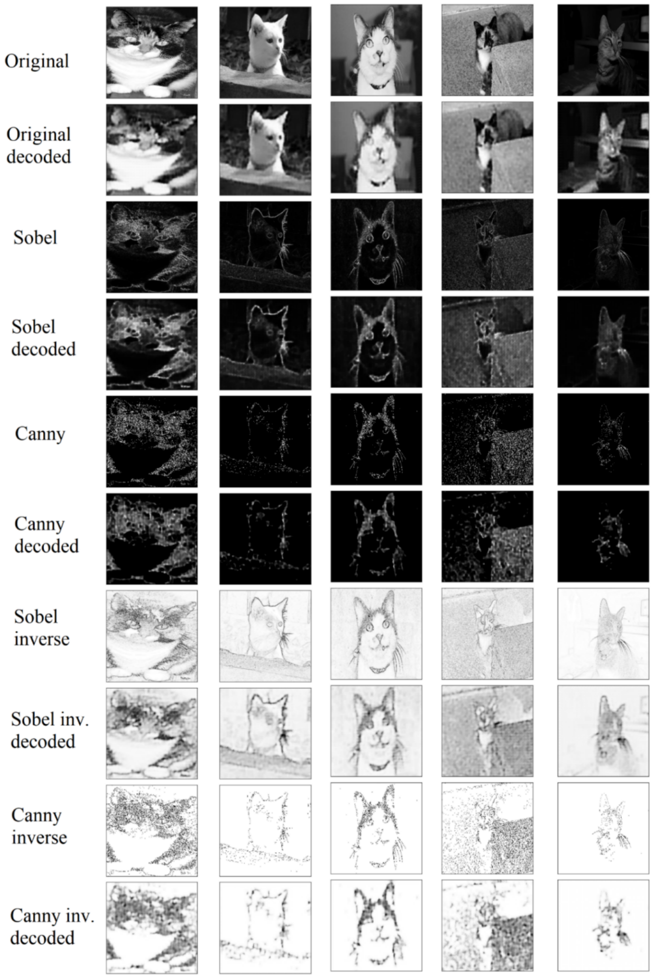


Figure 11. Comparison between the autoencoding of the Sobel and canny filtered images with both of their versions

For the sobel-direct, the mean and standard deviation values were 0.202 and 0.044, respectively. Their inverse scores were 0.699 and 0.124, respectively. For the canny-direct, the mean and standard deviation values were 0.234 and 0.090, respectively. The inverse scored 0.741 and 0.150, respectively.

These values support our first observation, which is that the autoencoder learns faster when the image's most important features are darker than the rest of the data. The experiments so far have resolved into using the Sobel and Canny filters, and more specifically, their inverted results. At the start, it was thought that the experiments would resolve into choosing only one filter as a preprocessing stage for the autoencoding, but as calculated previously, the quality of images between the Sobel and Canny images is so close that it does not imply the disregard of one of the two filters.

Nevertheless, there is a significant drop in quality when applying a high-pass filter to the original image and then passing it through an autoencoding stage. This raised a flag that perhaps the pipeline's order might not be thorough. For

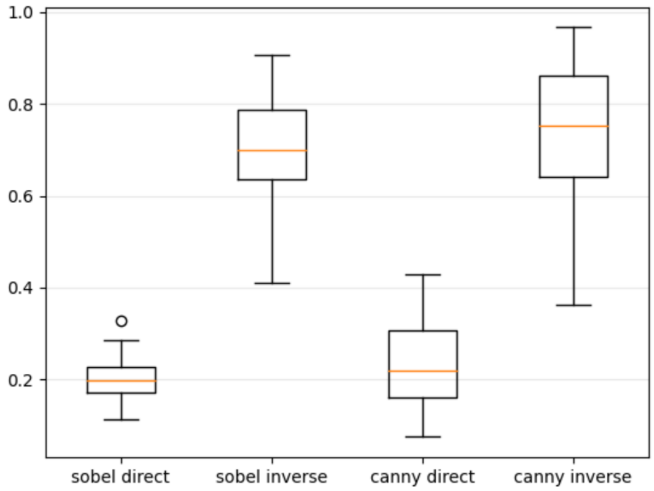


Figure 12. SSIM of different autoencoding approaches

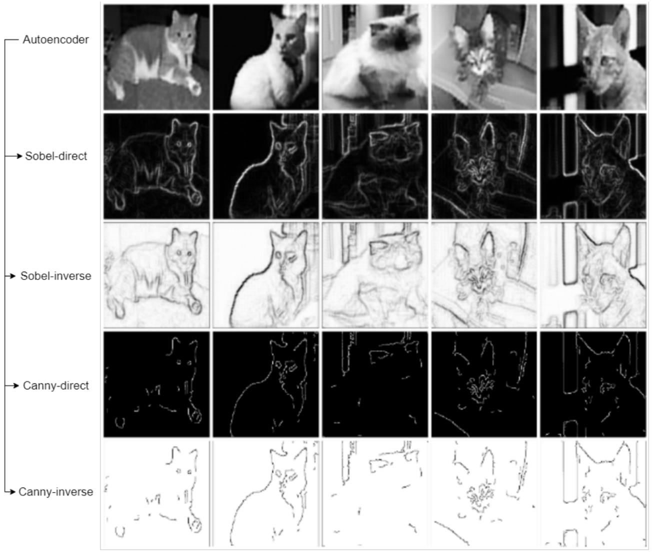


Figure 13. Filtered autoencoder images with Sobel and canny (both versions each)

instance, the autoencoder is perceived to work as a reconstruction algorithm. Simultaneously, it can be considered to smooth the image, or in other words, to represent it with more coherence between the pixel values. As a result, the high-pass filters may be more efficient if applied after image reconstruction rather than before autoencoding, which appears to cancel out some of the emphasis generated by the filters. Hence, an experiment on the matter should be performed.

C. Autoencoders as a Preprocessing Stage to High-Pass Filters

In this experiment, random images were taken, reconstructed with an autoencoder, filtered, and then vectorized. This experiment aims to display the effect of high-pass filters on reconstructed image vectorization. The five random resulting images are shown in Figure 13.

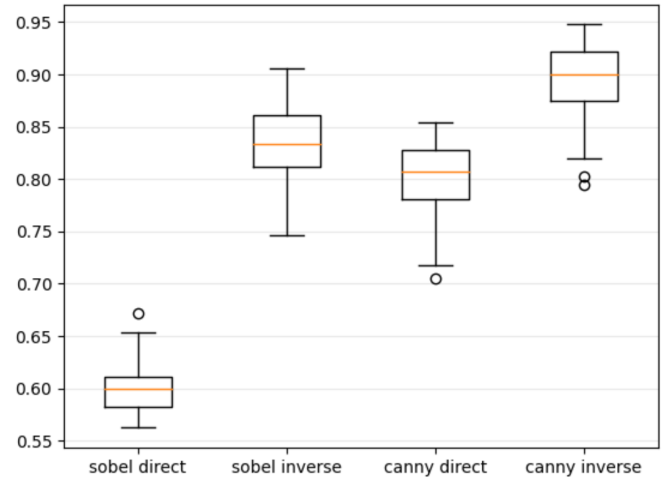


Figure 14. SSIM comparison of the vectorization of each of the four groups of images

The first impression the experiment gives off is that the filters brought more definition to the lines in the images, which made the shapes appear clearer. This can lead to better vectorization, as it depends on the definitions of the shapes represented in the tags.

However, there are two versions of each of the two filters, which suggest an evaluation of the vectorization of each of the four result groups. Therefore, an SSIM calculation was done between every filtered image and its vector format in a pool of 50 images, randomly selected. The results are displayed in Figure 14.

The box plots show the better fitness of white images with black lines when compared to the darker images in vectorization. Visually speaking, the Sobel filter results were more recognizable to the naked eye. However, it left more complexity in the image, which made it harder for the vectorization to be more exact. Therefore, it is concluded that the darker shapes are going to be used in both filters, while there is not yet a clear endpoint to resolve depending on only one of the two filters. Hence, a parallel stage of execution is introduced, which takes the autoencoder images and filters them with one filter before passing them to the global vectorization stage.

VI. Evaluation

Evaluation is concerned with how abstract the resulting images are. As there are two pre-processing blocks (filtering and autoencoding), four different pipelines can be built: autoencoding, filtering, autoencoding-filtering, and filtering autoencoding. After one of these selections is fed the images, a vectorization process is always cascaded at the end.

First, all of the resulting images are going to be evaluated based on their path count (size) and similarity to the input images. Then, a summary of the evaluation is going to be introduced for each of the pipelines individually.

Before engaging in the evaluation, it is good to elaborate on the column naming of the upcoming plots:

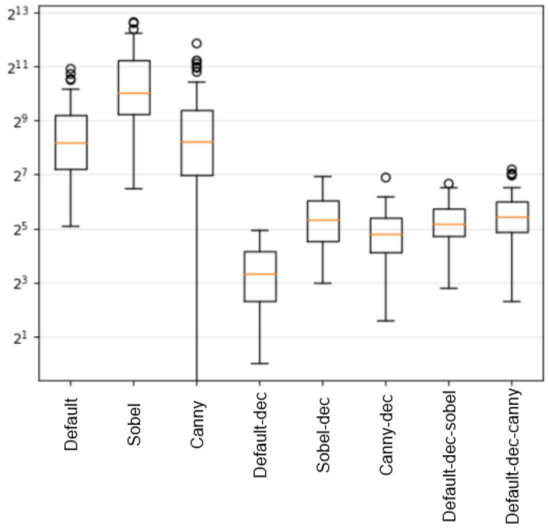


Figure 15. Path count of the resulted groups of vector images

- default: the default image.
- Sobel, canny: the filtered version of the image by the respective filter.
- dec: the decoded version.
- vect: the vectorized version.
- A combination of two or more indicates the case of cascaded stages. A default-dec-sobel label represents the following: the default image is reconstructed with the autoencoder and then filtered with the sobel filter.

A. Evaluating the size of the produced images

To evaluate the size of the image, we count the number of path objects generated in the SVG file. From Figure 15 (note that the graph is in logarithmic scale) we see that the autoencoder (*-dec-*) significantly reduced the size of images, as it keeps only the most important features. The reconstructed filtered images (canny-dec, sobel-dec) had a similar path count. Although it was much smaller than the ones that did not go through that step, it was still above the default images that were reconstructed and vectorized without any filtering. Finally, when filters were applied to the default images that were put through an autoencoding stage (default-dec-sobel, default-dec-canny), these images scored in size calculations very similarly to the filtered images when only reconstructed (canny-dec, sobel-dec).

B. Evaluating the quality of the produced images

A more accurate way of examining the efficiency of the vectorization process of each pipeline is to compare the images and their vector versions (Figure 16). The pipeline of autoencoding-filtering-vectorization (two last groups on the most-right) seems to experience the highest SSIM, which indicates its fitness in vectorization. It made more sense for the autoencoder to reconstruct the images and then for the filters to come afterward, emphasizing the important features of each image.

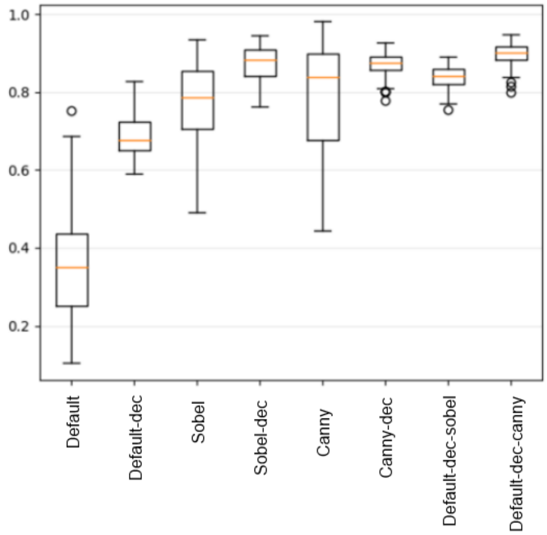


Figure 16. Vectorization accuracy of different pipelines

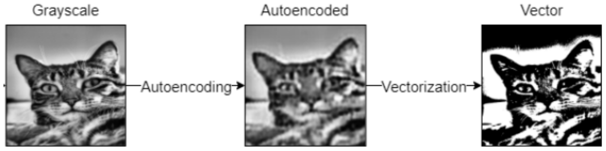


Figure 17. Autoencoding-vectorization pipeline

C. Implemented Pipelines: an evaluation summary

This is a summary of the evaluation of the results for each of the pipelines individually.

- **Autoencoding-Vectorization:** This pipeline was based on the work of Fischer and Amesberger [12]. However, the implementation was different, and the evaluation was about the abstractness of the results. The quality of the vectorization is acceptable only in terms of general similarity. However, an abstract representation of the image is not achieved (Figure 17).
- **Filtering-Vectorization:** In this pipeline (Figure 18), the vectorization algorithm finds difficulty in vectorizing the filtered images. This is due to the noises caused by the applied filters. Although the experiments showed that the quality of the vectorization increased when the images were taken as a light background with dark features, the noise involved created an obstacle for Potrace to convert thoroughly the images into a vector format, which resulted in losing data.
- **Filtering-Autoencoding-Vectorization:** This pipeline was built as an attempt to enhance the *Autoencoding-Vectorization* pipeline. Although the autoencoding stage was efficient in reducing the size of the images, it did not result in an abstract view of the image features. Therefore, a filtering stage was placed before the autoencoding process. Unfortunately, this pipeline does not achieve the result intended. The autoencoding stage was supposed to

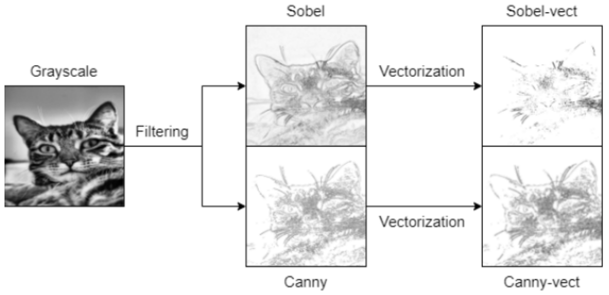


Figure 18. Filtering-vectorization pipeline

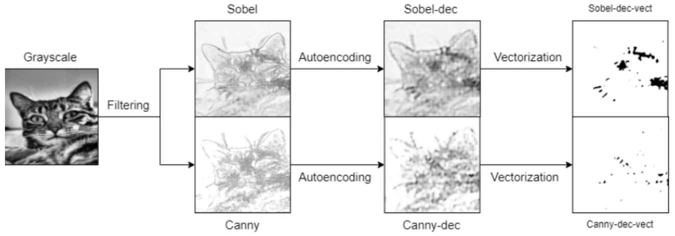


Figure 19. Filtering-autoencoding-vectorization pipeline

reconstruct the filtered images in a lower complexity; but the case at hand is that the autoencoding model is attempting to smooth the images, canceling the effect of the high-pass filters. This has resulted in a significant drop in the quality of the vector images, which is seen in Figure 19.

- **Autoencoding-Filtering-Vectorization:** Due to the results in the *Filtering-Autoencoding-Vectorization* pipeline, it was clear that the filtering stage would act more appropriately if it succeeded the autoencoding process, rather than preceding it. This was concluded when the autoencoding model was seen to reduce the complexity of the images while introducing a smoothing effect. The filters were placed after the reconstruction stage to preserve the important features of the reduced-complexity image. This cascade shows an acceptable vectorization quality while resulting in the intended abstract representation of the images as shown in Figure 20.

As for providing more visualizations of the results that can be obtained with this pipeline, Figure 21 shows some random images that were fed to the Autoenconding-filtering-vectorization pipeline along with their respective output images. As can be seen, the features of the cats are extracted very clearly in all examples.

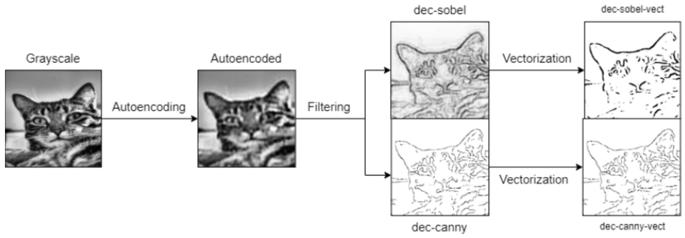


Figure 20. Autoencoding-filtering-vectorization pipeline

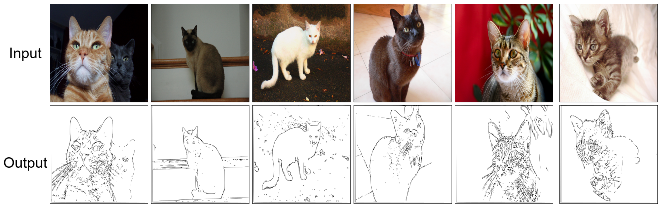


Figure 21. Some of the output images along with their input images of the pipeline Autoenconding-filtering-vectorization

VII. Conclusion

This paper discusses the autoencoding step and the use of high-pass filters in vectorization pipelines. As demonstrated, high-pass filters can improve the training of an autoencoder, which in turn improves the efficiency of vectorization by maintaining key aspects of an image.

The images that underwent the cascade of autoencoding-filtering scored the greatest in similarity and the lowest in error after the vectorization algorithm's effectiveness in each pipeline was assessed. This indicates that the most crucial elements of the reconstructed images were maintained and that the filtering step that came after the reconstruction enhanced those features even further, resulting in a better vectorization and a more abstract representation of the image.

Although the results from this cascade of autoencoding-filtering were respectable and met the initial objectives, more work needs to be done on the training dataset and model structures.

Regarding future work, experiments showed that dark features on a light background in images can improve both the training of autoencoder models and the process of vectorization. This will be an issue for further investigation. As this paper deals with single-channel (i.e., gray-scale) images, another aspect of the investigation will be the vectorization of multi-channel images.

References

[1] A. Fischer and Z. Bettouche, “High-pass filters preprocessing in image tracing with convolutional autoencoders,” in *IARIA, Copyright: Copyright (c) The Government of Germany, 2022. Used by permission to IARIA. ISSN: 2308-4170, ISBN: 978-1-61208-954-6*, Barcelona, Spain; April 28, 2022.

[2] P. Selinger, “Potrace : a polygon-based tracing algorithm,” in *Potrace*, 2003.

[3] N. Kanopoulos, N. Vasanthavada, and R. L. Baker, “Design of an image edge detection filter using the sobel operator,” *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[4] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[5] S. Lu, W. Jiang, X. Ding, C. S. Kaplan, X. Jin, F. Gao, and J. Chen, “Depth-aware image vectorization and editing,” *Vis. Comput.*, vol. 35, no. 6–8, p. 1027–1039, jun 2019. [Online]. Available: https://doi.org/10.1007/s00371-019-01671-0

[6] A. Bera, “Fast vectorization and upscaling images with natural objects using canny edge detection,” in *2011 3rd International Conference on Electronics Computer Technology*, vol. 3, 2011, pp. 164–167.

[7] K.-W. Chen, Y.-S. Luo, Y.-C. Lai, Y.-L. Chen, C.-Y. Yao, H.-K. Chu, and T.-Y. Lee, "Image vectorization with real-time thin-plate spline," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 15–29, 2020.

[8] M. Bessmeltsev and J. Solomon, "Vectorization of line drawings via polyvector fields," 2018.

[9] V. Lacroix, "Raster-to-vector conversion: Problems and tools towards a solution a map segmentation application," in *Raster-to-Vector Conversion*, 03 2009, pp. 318 – 321.

[10] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[11] Z. Gong, L. Xu, Z. Tian, J. Bao, and D. Ming, "Road network extraction and vectorization of remote sensing images based on deep learning," in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2020, pp. 303–307.

[12] A. Fischer and M. Amesberger, "Improving image tracing with artificial intelligence," in *2021 11th International Conference on Advanced Computer Information Technologies (ACIT)*, 2021, pp. 714–717.

[13] W. Zhang, J. Sun, and X. Tang, "Cat head detection - how to effectively exploit shape and texture features," in *ECCV*, 2008.

[14] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

[15] F. Chollet *et al.* (2015) Keras. [Online]. Available: https://github.com/fchollet/keras

[16] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[17] C. Sammut and G. I. Webb, "Mean squared error," *Encyclopedia of Machine Learning*, no. 4, pp. 653–653, 2010.

[18] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[19] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

An In-depth Comparison of Experiment Tracking Tools for Machine Learning Applications

Tim Budras*, Maximilian Blanck†, Tilman Berger†, and Andreas Schmidt*‡,
* Department of Computer Science and Business Information Systems,
Karlsruhe University of Applied Sciences
Karlsruhe, Germany
Email: {buti1021, andreas.schmidt}@h-ka.de
† inovex GmbH, Karlsruhe, Germany
Email: {mblanck, tberger}@inovex.de
‡ Institute for Automation and Applied Computer Science
Karlsruhe Institute of Technology
Karlsruhe, Germany
Email: andreas.schmidt@kit.edu

***Abstract*—As the machine learning market is growing strongly and machine learning is increasingly being used productively, new challenges for developers and operators arise that haven't been existing in traditional software development. One of these challenges is the versioning and reproducibility of models. To help solve this challenge *experiment tracking tools* exist, which keep track of the experimental development process of machine learning models. This paper describes the process of bringing a machine learning model to production and emphasizes its experimental nature and the challenges arising with it. Following the definition of a set of requirements for experiment tracking tools, 20 tools found in a market research are presented. Four of those tools are analysed in-depth, showing that differences between tools exist especially for advanced requirements. This paper also includes the progress the tools have made within the last year.**

***Index Terms*—Machine Learning; Experiment Tracking; Development Environment; MLOps**

I. Introduction

This paper is an extended version of a conference paper [1], published in 2022 at the Fourteenth International Conference on Advances in Databases, Knowledge, and Data (DBKDA-2022) conference in Venice/Italy. In this extended paper, we go into more detail about the various tools that we were able to consider in the previously mentioned conference paper. We have also examined the tools in terms of their current enhancements.

The machine learning market is growing strongly. According to MarketsandMarkets [2], it is ”expected to grow from USD 1.03 billion in 2016 to USD 8.81 billion by 2022”. As a result of this growth, tools have been developed in recent years to help develop machine learning models and put them into production. However, due to the fact that the use of machine learning in productive software is relatively new, tools and conventions are less settled and less commonly applied than in traditional software development.

Warden [3] uses the term ”machine-learning-reproducibility-crisis” to describe that the tools to meet these needs are often not deployed or used in practice. With regard to tracking data, parameters, models and results, numerous products with different focuses and strengths have been developed. Tools that focus on saving information around the model training and development process are often referred to as *experiment tracking tools*. But as stated in a Kaggle survey [4], in a large amount of scenarios these relatively new tools remain unused and tracking is either done manually or not done at all.

But without experiment tracking, the information under which circumstances an AI model was created is missing. It is therefore a black box model, which contradicts the High Level Expert Group on Artificial Intelligence (HLEG AI) [5] demand for *transparency*. An important criterion according to the HLEG demand for transparency is *explainability*: decisions made by an AI system must be understandable and comprehensible to humans. This requires information about the underlying datasets, the algorithms, parameters and data processing pipelines used, and the results obtained, which then serve as the basis for selecting specific algorithms/parameter sets.

The information gathered in this way will further enable repeatability of the experiments for both the current and future development teams, and also for other working groups dealing with the same or similar issues. Experiments by Alahmari et al. [6] show that the tracking information collected is sometimes not sufficient for the repeatability of an experiment. They demonstrated that running the same experiment several times can lead to different results. Reasons for this are, for example, a random selection of the training and test data, different libraries used or also hardware. In [7], for example, it was shown from Nagarjan et al. that when switching from CPU to GPU, different but deterministic results were obtained for the respective processor unit. More information about reproducibility and traceability to achive trustworthy AI can be found in [8], [9].

The paper is structured as follows: In Section II we explain the *machine learning lifecycle* and what artifacts, i.e., code, data, environment, parameter settings need to be tracked in the context of an experiment. Based on these findings we present in Section III the general architecture for experiment tracking tools and formulate the most important requirements. In Section IV four tools are presented and compared in detail. The paper is finished with a conclusion and outlook to further research directions in Section V.

II. Background

In this section, a set of basic insights required for understanding tracking tools in the field of machine learning will be presented as well as information about research related to experiment tracking tools.

A. The Machine Learning Lifecycle

The different phases and steps around the productive use of a machine learning model have been described by different authors using different terms. One of these terms is the *machine learning lifecycle*. Garcia et al. [10] describe the machine learning lifecycle as a three-phase process as shown in Figure 1.

The first phase is the *pipeline development*. During this iterative phase, the data preprocessing, exploration and visualization is done, model designs are chosen and models get trained with different configurations and hyperparameters. The authors emphasize that the important achievement of the first phase is not the model, but the pipeline that can be reused to create a model from a dataset. This pipeline can be used later in the second phase *training* (Figure 1, middle), to train and validate the model used for inference. The last phase (Figure 1, right) is called *inference*. Here, the prediction service (which includes the data preprocessing as well as the model used for inference) returns a prediction for a given user input. This service provides information about the predictions made, which can be used for subsequent training. The authors mention that the different stages are often managed by different teams.

Amershi et al. introduce a similar process, the *machine learning workflow* [11]. This process is divided into nine stages and is shown in Figure 2. Those nine stages can be grouped into four phases. The first phase consists of the model requirements stage, in which the objective of the machine learning task is defined. Furthermore, the type(s) of model that could be used to implement this objective get selected. The initial planning phase is followed by the data preprocessing phase, which includes the stages data collection, data cleaning and data labelling. The next phase describes the model engineering and includes potential feature engineering, as well as the model training and evaluation. If a good performing model is found, the model can be deployed into production, in the last phase, the deployment and monitoring stages in Figure 2. Once the model is used in production, the model needs to be monitored, to measure its performance and find out at which point a potential retraining is needed. The authors emphasise that – in contrast to their illustration (Figure 2) – the machine learning workflow is generally not linear, as the illustration implies. The workflow is iterative and contains multiple feedback loops, as indicated by the arrows.

B. Experiment Tracking

Langley [12] describes machine learning as an experimental science and compares the process of finding a good model to the empirical sciences of physics and chemistry. This aligns with the results from interviews Hill et al. [13] conducted with various machine learning practitioners in 2016. Seven out of seven interviewees experienced the need ”to resort to basic trial and error”. Langley defines an experiment as the process of examining the effect of varying one or more independent variables on some dependent variables [12]. Hence, an experiment consists of multiple runs. According to Vartak et al. [14], ”data scientist often built hundreds of models before arriving at one that met some acceptance criteria”. Each model built can be seen as the dependent variable of a run. However, experiment tracking tools can also be used in the pipeline development phase, introduced by Garcia et al. [10], which does not produce a model, but a training pipeline. In this case, the dependent variable would be the training pipeline. Therefore, the following definition of an experiment is used in this paper:

Definition (Experiment): A run is a part of an experiment, it has a specific set of independent variables that produces a model or a training pipeline. An experiment is a collection of runs that try to solve the same problem or business task. The objective of an experiment is to find the set of independent variables that results in the best dependent variable(s).

It should be noted that usually in practice it is not possible or at least not economically feasible to find the best independent variables [15].

Various possibilities exist to assess the quality of a model. A common approach is to calculate a metric for prediction quality (such as accuracy) on a dataset not used for training. However, additional (nonfunctional) quality measures might exist, e.g., the inference time, the training time or the explainability of a prediction.

Due to the fact that the number of experiment runs might be enormous, it is very helpful to track the experiment and its runs. The term *experiment tracking* describes the process of saving the information related to the experiment and its runs, to allow further evaluation. Although typically the verb *to track* is used in combination with experiments, some tools evaluated in this work have functionalities that use the words *log* or *logger*. Thus, both terms are treated as synonyms in this work.

In its easiest version, tracking can be done manually, alternatively one of the tools presented in Section IV can be used. A Kaggle survey conducted in 2020 [4, p. 27] showed that in most cases tracking is either done manually or not done at all. In theory, automating the process of tracking by using one of the tools presented later should have many

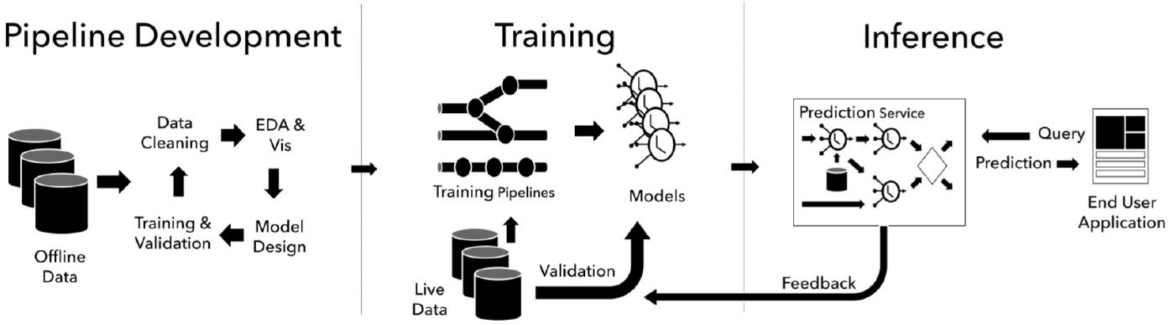


Fig. 1. Machine Learning Lifecycle (from [10])

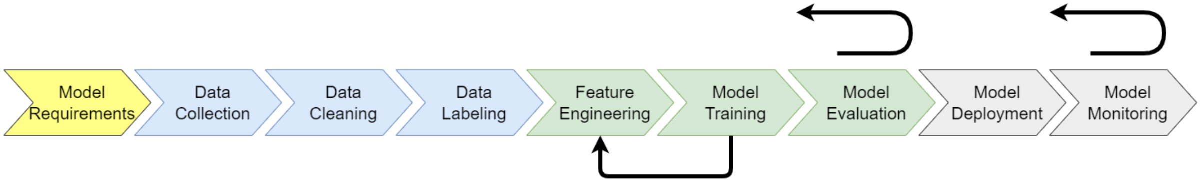


Fig. 2. Machine Learning Workflow based on [11]

advantages. Manual tracking can be error prone, wrong data may get saved or the tracking may be forgotten. Additionally tools might provide functionality to facilitate working in teams or analyzing the tracked data. Automating the process allows the machine learning practitioner to focus on developing the best model.

Either way, tracking experiments brings multiple advantages: Keeping track of all the runs makes it easy to find the best variables. Additionally, it is easy to see, which sets of independent variables have already been tried out or might be worth trying out in the future. This is especially helpful if the work is done in teams, or if the responsible person changes. With the right tool, tracked experiments can be easily compared. If a model is used in production, it can be very helpful to have the information available on how the model was created. Another advantage – which applies especially to research – is the fact that results may need to be reproduced. Furthermore, establishing the use of an experiment tracking tool in a company or a project provides the benefit of a structured way to access the data generated during experimentation, regardless of the individuals responsible for the experiments.

C. Related Work

Experiment Tracking can be seen as a part of MLOps, which can be described as a common set of practices that includes the development of a machine learning application as well as its operations [16]. 41% of business decision makers named ”versioning and reproducibility of models” as a machine learning challenge in a survey conducted by Algorithmia in 2020 [17], making it the second most often named challenge after ”scaling up” with 43%. Experiment tracking should help to tackle this challenge.

Research has been conducted and lead to the presentation of individual frameworks such as MLflow [18], [19]. Additionally, new tools have been developed or proposed based on evaluations of existing tools by Scotton or by Zárate et al. [20], [21]. Other work e.g., by Hewage and Meedeniya focuses on comparing existing experiment tracking tools [16]. The work by Hewage and Meedeniya is different to our work, as it does not include a comparison about the ease of use nor the accessibility of tracked data. Also, the selection of compared tools is different. It does not include DAGsHub or Neptune but includes other tools instead.

D. Reproducibility Requirements

In a reproducibility challenge, Pineau showed that most challenge attendees found it at least reasonably difficult to reproduce the result of a paper of the *International Conference on Learning Representations 2018* [22]. Pineau also published a machine learning reproducibility checklist [23], which is supposed to help increase the reproducibility of experiments. Tatman et al. [24] define three levels of reproducibility for research: low, medium and high reproducibility. The lowest level of reproducibility is achieved by publishing the paper. According to the authors, the medium level is achieved, when the code is published along with the used data. The highest level can be reached by additionally providing the environment.

In the following subsections the requirements for reproducibility introduced by Tatman et al. [24] as well as the terms *hyperparameters* and *metrics* will be explained in detail.

1) Code: Similar to traditional programming, machine learning highly depends on the source code. There are several tools to effectively version source code. A developer survey by StackOverflow in 2018 [25] showed that almost 90 % of the developers use Git as a version control system. There is no valid reason to not track the code used in machine learning projects with Git. However, in a fast developing process, experiment runs might be executed, without committing the code beforehand. This would lead to a lack of reproducibility, as Git needs a commit to restore a state of the code.

2) Data: Besides the code, data plays an essential role in machine learning, because different data can lead to different results. As the kind of data depends on the business task, the data format varies. Common data formats are text, image or video. Due to the partly large data resources, a suitable tool for the efficient storage of different variants of a data resource should be used.

3) Environment: Providing information about the environment is certainly only necessary for some use cases. However, it can contain important information of the original run, such as the used hardware, the used operating system or the software dependencies. Thus, keeping track of the environment can be helpful to reproduce a run. Tatman et al. [24] propose three possibilities to share the environment: Either by using a hosted service, or by providing a container or virtual machine, which includes all dependencies. At minimum, the used libraries and their versions should be tracked.

4) Hyperparameters: According to Bergstra et al. [26], hyperparameters configure the machine learning algorithm before training, whereas, in the present paper, any kind of configuration parameters of the experiment run (not only the machine learning algorithm) will be considered as hyperparameters. As any change in configuration might result in different results, it is recommended to track as many hyperparameters as possible. Although hyperparameters are often tracked implicitly when they are defined in the code and the code is versioned, hyperparameters should be tracked explicitly to allow easier comparison.

5) Metrics: A metric is an evaluation measure calculated to quantify "the effectiveness of a complete application that includes machine learning components" [15]. Most of the times, metrics will be calculated based on a model's predictions on data that has not been used for training. Different metrics with varying strengths and weaknesses exist. For classification tasks for example, accuracy or precision can be used. Accuracy is defined as the fraction of correct predictions out of all predictions [15]. Metrics can be used to compare different runs of an experiment and can be considered as one of the dependent variables of the experiment. Whatever type of metric is used is actually not important for experiment tracking.

III. Experiment Tracking Tools

The main goal of experiment tracking is to save information during experimentation in order to be able to access it later. As a result, most experiment tracking tools consist of at least three components, as shown in Figure 3. Some kind of client software – for example a Python library – is required to store the tracked information during experimenting on a persistent data storage or send it to a server. The data can often be retrieved programmatically through the client or be viewed in a Graphical User Interface (GUI). The exact functionality of those components differs between the available tools.

A. Requirements

As already discussed in Subsection II-B, tracking of code, data, the used environment, hyperparameters, and metrics are elementary requirements for such a tool. Additional requirements examined in our research also include the following aspects:

1) Storing of Models: Training a model can take a long time. Therefore, the models should be stored and linked to the hyperparameters and metrics. This avoids time consuming retraining e.g., if a model should be evaluated on new data.

2) Accessibility of Tracked Information: Tracking is a prerequisite however, the tracked data will only provide value, if the tracked information can be accessed in a simple yet powerful way. This includes a user interface, which provides a clear and customizable overview of all runs, as well as the possibility to compare runs in depth. Filtering the runs with easy but rich querying options is also part of this requirement. Besides that, the tool should provide a possibility to create and show plots. If additional interfaces, e.g., an API, exist, they will be useful as well.

3) Collaboration: According to Tabladillo et al. [27], bringing data science projects to production requires different tasks. For this reason, data science projects are often worked on in teams composed of different roles. Therefore, the tool should facilitate collaborative work. This includes the possibility of viewing existing results of different team members and adding new results by executing new runs. To achieve this, a form of access management is required.

4) Initial Setup and Infrastructure: Because tracking machine learning experiments should facilitate the work of teams, tools will only be taken into consideration if they have low barriers to entry. Thus, this requirement describes the initial investment needed to set up and use the tool. The initial setup is everything that does not need to be repeated if the same tool is used in another project (given the projects can use the same infrastructure). As cloud tools might have an advantage concerning the initial setup, it must be kept in mind that saving data off-premises might not be a possibility in any case due to legal or corporate regulations.

5) Ease of Integration: Similar to the previous requirement this requirement concerns user-friendliness. Yet, unlike the initial setup and infrastructure, the ease of integration describes how easy it is to include the tool into a specific project. This means, for example, project-specific configuration or source code changes.

IV. Examined Tools

In a market research, the following tools with experiment tracking functionality were identified.

- Aim [28]
- Amazon SageMaker Experiments [29]
- Azure Machine Learning [30]

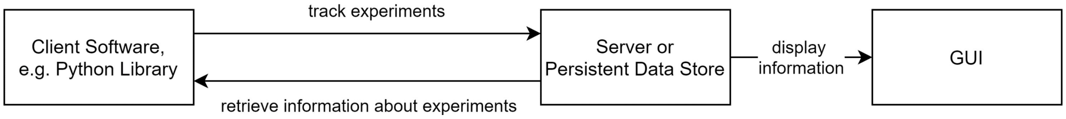


Fig. 3. General Architecture of an Experiment-Tracking-Tool

- ClearML [31]
- Comet [32]
- DAGsHub [33]
- DominoDataLab [34]
- Guild AI [35]
- H2O MLOps [36]
- Iterative Studio [37]
- MLflow [38]
- Neptune [39]
- Paperspace Gradient [40]
- Polyaxon [41]
- Sacred [41] in combination with Omniboard, Incense or Sacredboard (GUIs)
- TensorBoard [42]
- Valohai [43]
- Verta [44]
- Vertex AI [45]
- Weights & Biases [46]

The research was conducted online, using search engines, blogs, forums, as well as the websites of the respective tools.

To allow an in-depth evaluation of the tools in the scope of this work, the tools listed previously have to be limited to a reasonable amount. The tools were selected in consultation with a project team at inovex, actually developing a multilingual and multidomain Conversational AI. The selection was influenced by requirements given from the project team. First, it was required that tracking tool is not running in a cloud ecosystem only nor creates a platform lock-in, like Azure Machine Learning, Amazon SageMaker Experiments or Paperspace Gradient do. Furthermore, the tracking tool should be independent of runtime and used libraries, which did exclude Tensorboard. For the sake of brevity four tools were examined only. The selection was based, in addition to the stars on github, on the simplicity of integrating the tools into the project and the familiarity within the project team. Therefore, Aim and Polyaxon were not considered, but they are very interesting tools and should be included in future research.

In this process, MLflow, ClearML, Neptune and DAGsHub were adopted for a more detailed evaluation. MLflow was selected because it is one of the most established and widely used tools. ClearML was assessed because of its wide range of operating options. It can be used for free (even in small teams) as a hosted option, operated self-hosted for free, but also be used with a paid plan. The most important argument for choosing Neptune was that it promises an effortless setup. The last option evaluated was DAGsHub, as it makes use of Data Version Control (DVC) [47] for versioning data, like the project. In the next subsections each tool will be evaluated based on the requirements defined in Subsection III-A and an exemplary integration will be provided.

A. MLflow

The open-source tool MLflow is developed by Databricks and was introduced by Zahari et al. [48] and launched in June 2018. At the time of writing this paper (October, 2022) the latest released version was 1.29.0 [49]. The software itself is shipped as a Python package and can be either hosted on own server or used as a Software as a Service (SaaS) offering by Databricks called *Managed MLflow*. Since Databricks' main business is offering managed versions of Apache Spark clusters, Managed MLflow is tightly coupled to these offerings. By comparing the managed with the open source version the managed offers mostly features that allow integrations into the Databricks eco-system. In contrast to the self-hosted version the managed one offers notebook and workspace integration to Databricks, as well as a role based user management. In addition also an integration to the aforementioned clusters is offered [50].

With the version 1.27.0 MLflow introduced a new experimental feature called *MLflow Pipelines*. This feature provides a framework to structure the whole cycle of an machine learning project. Hereby the user can specify pipelines either with Python code or by configuration files. The steps of these pipelines define parts like data preprocessing, splitting data into parts, evaluating trained models or storing models. MLflow provides templates that fit for common machine learning tasks. Since the pipelines are either defined by Python code or configuration files they can be easily stored in a repository like Git [51].

As already mentioned the open-source version of MLflow is shipped as a Python package and can be easily installed by any Python package manager like pip or conda. MLflow uses a naming similar to our definition in Subsection II-B where runs are grouped into experiments. The most basic setup of MLflow just uses a local file system to store experiment and run metadata. In order to get a more scalable setup it is advised to setup a relational database as a backend. Here MLflow is able to use a varity of databases like mysql, mssql, sqlite and postgresql. Apart from experiment and run metadata MLflow uses either a local file system or a cloud object storage (like AWS S3, Google Cloud Storage or Azure Blob Storage) to store artifacts like trained models. If not configured explicitly the metadata as well as the artifact data are stored into the local file system [52].

```
1  import mlflow
2  mlflow.set_tracking_uri("postgresql://postgres:
       postgres@172.3...")
3  mlflow.set_experiment("MyProject") #group runs
4  with mlflow.start_run() as run:
5      hyperparams = {"lr": 0.01,}
6      mlflow.log_params(hyperparams)
7      #Training placeholder, model stored in var model
8      mlflow.pytorch.log_model(model, "log_r",)
9      mlflow.log_metric("acc", 0.99)
10     mlflow.set_tag("performance", "best")
```

Listing 1. MLflow example code

To start tracking with MLflow, a run has to be started as shown in Listing 1. By using a context manager to start the run, the run will be ended automatically (line 4). MLflow differentiates between metrics and params; both can be logged to MLflow by using the respective function. With the *log_params* (line 6) function a set of values like hyperparameters describing the current run can be stored. This function takes all kinds of values, which can be stringified. In contrast the *log_metric* (line 9) function, where only numeric values can be passed. That function is used to keep track for evaluation metrics (like precision or recall) for one run. Both functions exists as singular to log one value, or as plural to log multiple values (here, a dictionary is passed, as the only parameter and the name and values of the dictionary will be used. In addition a run can be marked by using the *set_tag* (line 10) function. That function is intended to mark a run e.g., as the current best performing. Not shown in the listing is the *log_artifact* function, which can be used to store a local file attached to the current to MLflow. Grouping multiple runs together allows easy viewing and comparison in the GUI. This can be achieved by setting up an experiment (line 3).

As mentioned before, MLflow uses the local file system by default to store metadata. By passing an URI pointing to a database to MLflow these data will be stored there [52].

The MLflow GUI in Figure 4 shows all the hyperparameters and metrics in a clear table. Runs of the same experiment can be compared and metrics are automatically plotted. In addition to the GUI, data tracked with MLflow can be retrieved via Python, R, Java and REST APIs. MLflow does not provide a dedicated way to keep track of the data used for training. It does not automatically log information about the environment either. However, with *MLflow Projects*, MLflow wants the users to manually specify their environment [54]. This can be achieved by creating a conda yaml file or a docker image and structuring the project by providing entry points and default parameters.

MLflow can be used for free in teams, however, this requires shared data storage, which has to be set up by yourself.

B. Neptune

Neptune is a tool developed by Neptune Labs. It is described as a ”metadata store for MLOps” [39]. Neptune consists of a server (closed-source) and a client (Python & R packages, open-source). The first version of the Python package was released in March 2019 [55]. At the moment (October 2022), 0.16.9 is the newest version. In the last year, an R client has also been added [56]. With 0.9.0 (released end of May 2021), the API received a significant update, introducing a new and slightly different way to use Neptune, while maintaining backward compatibility.

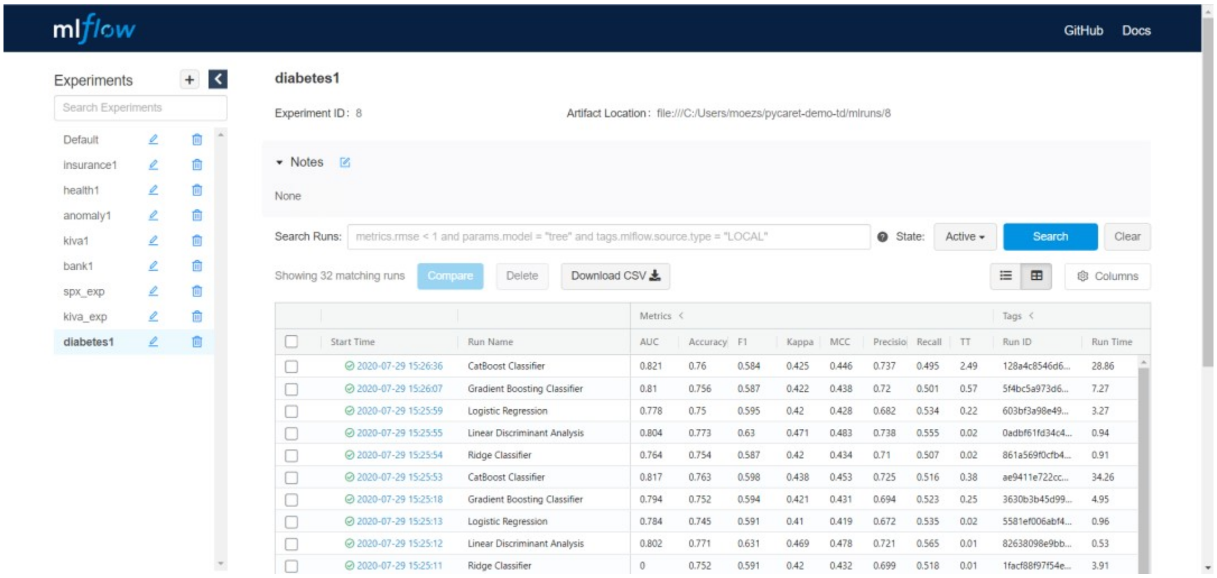
To get started with Neptune, an account has to be created at neptune.ai and an API token has to be generated. To track experiments, a project (similar to an experiment in MLflow) has to be created in the Neptune Web App or via the available management API. After those setup steps, Neptune is ready for use.

```
1  import neptune.new as neptune
2  run = neptune.init(project="tbud/MyProject")
3  hyperparams = {"lr": 0.01,}
4  run["data/train"].track_files("./datasets/train")
5  run["hyperparams"] = hyperparams
6      #Trainingloop placeholder
7      run["loss/train"].log(the_current_loss)
8  torch.save(model, "log_r.mdl")
9  run["model"].upload("log_r.mdl")
10 run["acc"] = 0.99
11
12 run["model_pickle"].upload(neptune.types.File.as\
       _pickle(model))
```

Listing 2. Neptune example code

Listing 2 shows the integration of Neptune, after importing the new Neptune API, we can initialize a run and assign it to a project (line 2). Neptune does not differentiate between metrics and hyperparameters. To log values with Neptune, a notation with square brackets and strings as keys (e.g., `run["some_key"]`) is used, which is similar to adding new values to a dictionary in Python (line 10). To track series such as the loss, the log function has to be used (line 7). This automatically generates a plot in the GUI. To structure values, a structured namespace can be used by putting a slash in the key name (e.g., `run["namespace/some_key"]`). This concept is then used to group and structure values in the GUI. The namespace can also be used to easily distinguish between metrics and hyperparameters. To upload a trained model, it first has to be saved locally (line 8) and can then be uploaded to Neptune using the upload method (line 9). Neptune provides a dedicated model registry that shows all production-ready models in a centralized way. It further enables the user to track transitions between different development stages of a model.

Neptune provides basic functionalities to keep track of the data used in the experiment runs. Neptune can calculate the hash value of a file or folder and store it with additional metadata (path, size and the last-modified date), which allows the user to see if the dataset has changed between experiment runs. This can be achieved by using the `track_files` method as shown in line 4. However, the dataset is not stored on the Neptune server and its data can not be retrieved. If a small dataset is used, it might be as well an option to upload the whole dataset. This will than be handled as an artifact, which is similar to the handling of a model file. A dataset can be uploaded by using the `upload()` function for single files or the `upload_files()` function for multiple files or directories. Arbitrary Python objects can be uploaded directly as a pickle (a Python-specific serialization format), without the need to locally store the pickle file first, by



diabetes1

Experiment ID: 8

Artifact Location: file:///C:/Users/moezs/pycaret-demo-td/mlruns/8

Notes

None

Search Runs: metrics.rmse < 1 and params.model = "tree" and tags.mlflow.source.type = "LOCAL"

Showing 32 matching runs

			Metrics <								Tags <	
☐	Start Time	Run Name	AUC	Accuracy	F1	Kappa	MCC	Precisio	Recall	TT	Run ID	Run Time
☐	2020-07-29 15:26:36	CatBoost Classifier	0.821	0.76	0.584	0.425	0.446	0.737	0.495	2.49	128a4c8546d6...	28.86
☐	2020-07-29 15:26:07	Gradient Boosting Classifier	0.81	0.756	0.587	0.422	0.438	0.72	0.501	0.57	5f4bc5a973d6...	7.27
☐	2020-07-29 15:25:59	Logistic Regression	0.778	0.75	0.595	0.42	0.428	0.682	0.534	0.22	603bf3a98e49...	3.27
☐	2020-07-29 15:25:55	Linear Discriminant Analysis	0.804	0.773	0.63	0.471	0.483	0.738	0.555	0.02	0adbf61fd34c4...	0.94
☐	2020-07-29 15:25:54	Ridge Classifier	0.764	0.754	0.587	0.42	0.434	0.71	0.507	0.02	861a569f0cfb4...	0.91
☐	2020-07-29 15:25:53	CatBoost Classifier	0.817	0.763	0.598	0.438	0.453	0.725	0.516	0.38	ae9411e722cc...	34.26
☐	2020-07-29 15:25:18	Gradient Boosting Classifier	0.794	0.752	0.594	0.421	0.431	0.694	0.523	0.25	3630b3b45d99...	4.95
☐	2020-07-29 15:25:13	Logistic Regression	0.784	0.745	0.591	0.41	0.419	0.672	0.535	0.02	5581ef006abf4...	0.96
☐	2020-07-29 15:25:12	Linear Discriminant Analysis	0.802	0.771	0.631	0.469	0.478	0.721	0.565	0.01	82638098e9bb...	0.53
☐	2020-07-29 15:25:11	Ridge Classifier	0	0.752	0.591	0.42	0.432	0.699	0.518	0.01	1facf88f97f54e...	3.91

Fig. 4. MLflow GUI (from [53])

using Neptune's file type with the `as_pickle()` method, as shown in line 12. In contrast to ClearML, Neptune does not automatically keep track of the computational environment. Thus, software dependencies are not saved when running an experiment, which makes reproducibility more difficult.

The GUI of Neptune looks similar to the MLflow GUI. It includes all the basic functionalities that MLflow has, but also has additional nice-to-have features, such as query completion for filtering or an option to save customized views. Figure 5 shows the run overview of the GUI. In the table, every run is represented by one row, the displayed columns represent metadata of a run and can be easily configured and even be renamed to a custom name. Neptune provides rich filtering options and helps the user writing the query by giving fitting proposals. Besides that, it is possible to group runs together to make the table clearer. Once a table view is customized as needed, it can be saved. This allows to quickly switch between various different views. Comparing multiple runs is easily done by clicking on the eye symbol for the desired runs. Neptune can filter for differences between runs and as well shows a small indicator to quickly see if a value increased or decreased.

Beside the GUI, the data can also be retrieved through a Python and R API. As a drawback, in contrast to MLflow and ClearML, Neptune does not provide a REST API.

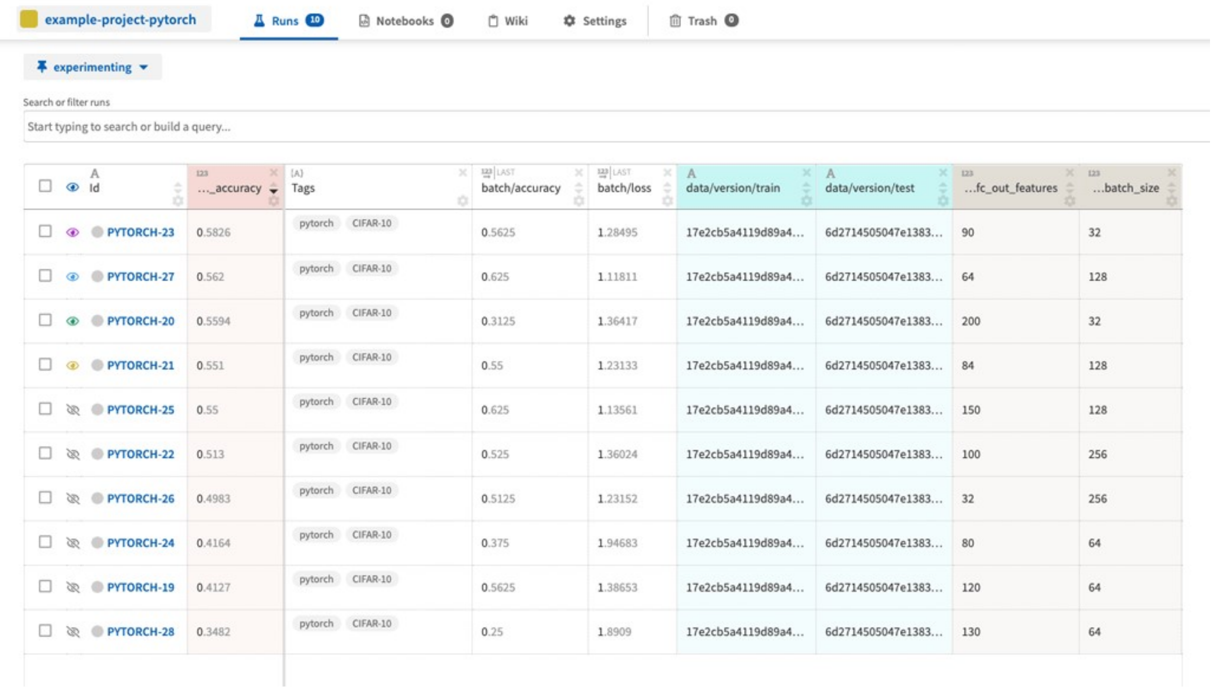
Similar to MLflow, Neptune's focus is tracking models, metrics and hyperparameters. Neptune has the opportunity to save all the tracked information on their servers, which is the most common and easiest way. However, this could raise data governance issues. Thus, Neptune does now also offer the possibility to deploy the server code on-premises or in a private cloud [57]. For single users and in special cases (e.g., academia, research) Neptune can be used for free. Paid plans have fixed prices, regardless of the amount of users. However, storage and usage limits exist (which can be increased by additional payments).

Neptune additionally provides the option to integrate Jupyter notebooks into projects. This can be done by installing the Neptune notebook extension. After enabling it, snapshots of notebooks can be created and saved to the Neptune project with the capability to compare different versions of the same notebook. This allows saving data exploration work next to the experiment runs. Furthermore, an external tool exists that allows the combination between Neptune and MLflow [58]. In this case, MLflow *runs* can be stored on a Neptune server. That way MLflow experiments can profit from the organization and collaboration features of Neptune. At the time of writing (October 2022), the new client wasn't yet supported by this tool. Also, Neptune provides a GitHub Actions template to make the data tracked in Neptune available in GitHub Pull Requests.

C. ClearML

ClearML is an open-source tool developed by Allegro AI, it was formerly known as Allegro Trains. At the time of writing, the current version of ClearML is 1.7.1. ClearML sends data to a ClearML server to store the data. This server can either be the SaaS solution provided by ClearML or a self-managed setup, which can be relatively easy set up for example with the Docker images provided by ClearML. While self operating is completely free, the free SaaS version has some limitations to it, such as the possible number of project members.

The ClearML client can be easily installed using pip. To use ClearML app credentials have to be added to the environment in which the experiment is conducted to connect the client with the account on the server. The credentials can be created in the workspace section of a ClearML account in the web interface.



Id	..._accuracy	Tags	batch/accuracy	batch/loss	data/version/train	data/version/test	...fc_out_features	...batch_size
PYTORCH-23	0.5826	pytorch CIFAR-10	0.5625	1.28495	17e2cb5a4119d89a4...	6d2714505047e1383...	90	32
PYTORCH-27	0.562	pytorch CIFAR-10	0.625	1.11811	17e2cb5a4119d89a4...	6d2714505047e1383...	64	128
PYTORCH-20	0.5594	pytorch CIFAR-10	0.3125	1.36417	17e2cb5a4119d89a4...	6d2714505047e1383...	200	32
PYTORCH-21	0.551	pytorch CIFAR-10	0.55	1.23133	17e2cb5a4119d89a4...	6d2714505047e1383...	84	128
PYTORCH-25	0.55	pytorch CIFAR-10	0.625	1.13561	17e2cb5a4119d89a4...	6d2714505047e1383...	150	128
PYTORCH-22	0.513	pytorch CIFAR-10	0.525	1.36024	17e2cb5a4119d89a4...	6d2714505047e1383...	100	256
PYTORCH-26	0.4983	pytorch CIFAR-10	0.5125	1.23152	17e2cb5a4119d89a4...	6d2714505047e1383...	32	256
PYTORCH-24	0.4164	pytorch CIFAR-10	0.375	1.94683	17e2cb5a4119d89a4...	6d2714505047e1383...	80	64
PYTORCH-19	0.4127	pytorch CIFAR-10	0.5625	1.38653	17e2cb5a4119d89a4...	6d2714505047e1383...	120	64
PYTORCH-28	0.3482	pytorch CIFAR-10	0.25	1.8909	17e2cb5a4119d89a4...	6d2714505047e1383...	130	64

Fig. 5. Neptune GUI (from [39])

```
1  from clearml import Task, Logger, Dataset
2  path = Dataset.get(dataset_project="MyProject/data",
        dataset_name="ds_1").get_local_copy()
3  task = Task.init(project_name="MyProject",
4          task_name="Task1", reuse_last_task_id=False,
5          output_uri="gs://MyProject")
6  hyperparams = {"lr": 0.01,}
7  task.connect(hyperparams)
8  cur_log = task.get_logger()
9  #Training placeholder, model stored in var model
10     cur_log.report_scalar("train", "loss", 1, ep)
11 torch.save(model, "log_r.mdl")
12 cur_log.report_single_value("accuracy", 0.99)
```

Listing 3. ClearML example code

An exemplary use of ClearML is shown in Listing 3. Initializing an object of ClearMLs Task class by calling its `init()` method, starts the tracking with ClearML, which is shown in line 3. Setting `reuse_last_task_id` to False (line 4) ensures that this task will not override an old task. The `output_uri` (line 5) specifies the location for the artifacts (e.g., the model) and is in this example set to a Google Cloud Storage. In ClearML a task is the name for everything that can be tracked, similar to a run in MLflow. By starting the tracking, ClearML automatically keeps track of a multitude of things, such as:

- Information about the Git repository, including the name of the current branch, the current commit ID and the output for the `git diff` command.
- Names and values of command line arguments that have been passed using standard Python packages, such as click or argparse.
- Plots created by libraries e.g., matplotlib, plotly or seaborn.
- Information logged by Tensorboard [42] and TensorboardX [59].
- Installed and used packages.
- Information about the resource usage (CPU, GPU, disk space, etc.).

Besides the information that is logged automatically, additional information can get logged with ClearML. This can be achieved by connecting an object to the task as shown in line 7. This object can be a Python dictionary or an object of a (custom) class.

An object of the ClearML class `Logger` is required, to log metrics. The `task.get_logger()` (line 8) and `Logger.current_logger()` (not shown in this Listing) functions return the logger object, which is is connected to the current task. To log metrics, the method `report_scalar()` of the logger object can be used as shown in line 10. This method is especially useful for metrics that change over time, as ClearML automatically creates a plot displaying the change over time in its GUI. The method requires four parameters: title, series, value and iteration. The title specifies the name of the scalar and the plot of the scalar. Multiple series can be grouped into one plot by providing the same title. Series describes the name of the series of the plot, value the value and iteration provides the x-coordinate for the plot line. For single values there is a `report_single_value()` method, which only requires the name and the value and does not result in the value being displayed in a plot. This is useful for reporting metrics that only have one value in an experiment run. ClearML also provides more sophisticated options such as `report_matrix()` to log a confusion matrix or `report_histogram()` to log a histogram.

Besides its hyperparameter and metric tracking capabilities, ClearML provides a possibility to efficiently store and manage large datasets. It works similar to DVC [47]: Before up- or downloading files, hash sums are calculated and compared to avoid traffic in case there have been no changes. ClearML also allows to store additional metadata about the data files, which can then for example be retrieved through the GUI. This allows versioning datasets even for binary files. A simple example of the integration into code is given in Listing 3. To get the local path to a dataset managed with ClearML, the dataset has to be queried with the `Dataset.get()` function (line 2). The `get_local_copy()` (line 2) function ensures that a local copy is available and returns the path, which can then be used for training. ClearML automatically tracks and uploads (trained) models if they are saved using the respective functions of the most common Python machine learning frameworks (e.g., Tensorflow, Pytorch, scikit-learn). The destination of the upload is specified during the initialization of the task (line 3), in this case a Google Cloud Storage, but other common cloud storages are also supported. The model can then be accessed in Python by retrieving the respective task and choosing the desired model.

As mentioned earlier, initializing a ClearML Task automatically tracks the installed and used packages including their version numbers. This can help to reproduce results at a later stage.

Figure 6 shows a screenshot of the GUI. Multiple tasks are collected in a project, ClearML also allows the creation of sub-projects for projects that require more organization. While the overview table of the experiments looks similar to Neptune and MLflow, the detailed view of the task is very nested and can overwhelm new users. This is in our opinion the biggest downside of ClearML compared to the other tools: due to its huge amount of possibilities, it requires more time to familiarize. However, we think this time is well invested since ClearML provides a lot of options and possibilities for the user. Those options include possibilities to show and hide columns as well as sorting and filtering or a function to compare runs, in which case the differences between runs will be highlighted. Besides examining data in the GUI, the data tracked with ClearML can also be retrieved trough the Python API or through a REST API. Projects are organized in workspaces, team members can be added to a workspace to allow collaborative work. In the free hosted version, workspaces are limited to three members, no such limit exists on the self-managed version. ClearML provides also additional features, such as logging debug samples of images, audio, video samples, which can help understanding the data, which was used to conduct experiments. To help increasing reproducibility, by default ClearML automatically sets a random seed for Tensorflow, Pytorch, and random.

D. DAGsHub

In contrast to the other presented tools, DAGsHub pursues a different approach. It makes use of existing open-source technologies and provides unified storage and a GUI for them (however, DAGsHub itself is not open-source). The open-source tools combined by DAGsHub are:

- DVC [47] is used to keep track of the data and models.
- Git keeps track of the code.
- MLflow or the DAGsHub Client can be used to track hyperparameters and metrics.

The interaction of the different tools is presented in Figure 7.

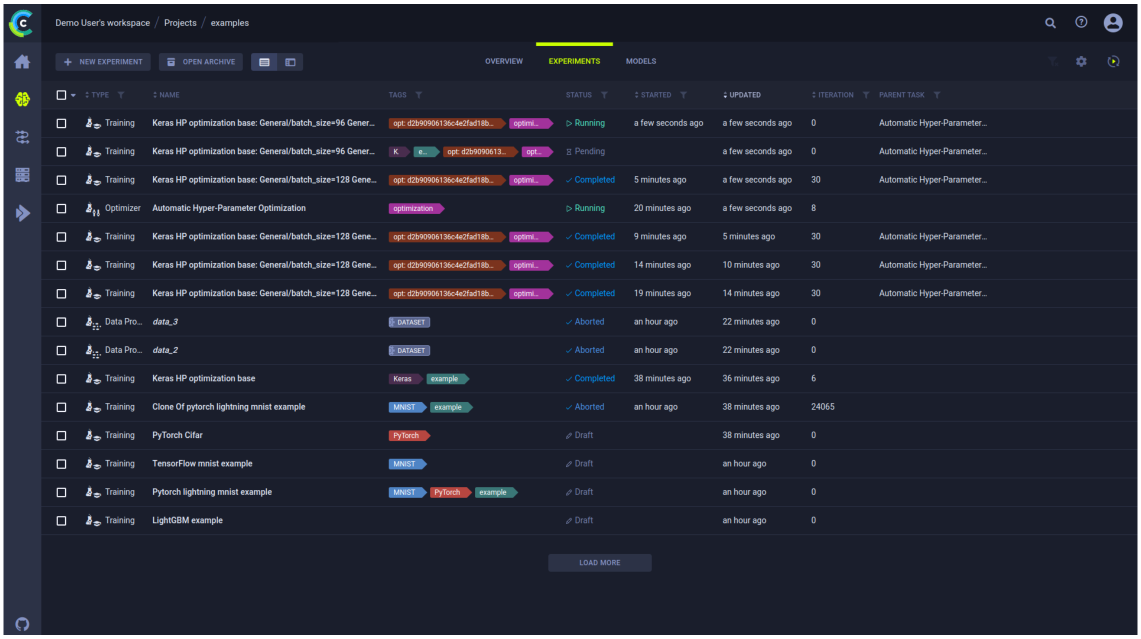
In order to take full advantage of DAGsHub, Git and DVC as well as MLflow or the DAGsHub Client should be installed on the client. In case of using MLflow to log to DAGsHub the integration into code is almost identical to the one shown in Listing 1. The only required change is to set the tracking URI specified in line 2 to the URI provided in the DAGsHub GUI. The functionality when using DAGsHub concerning the tracking of hyperparameters and metrics is similar to MLflow. An advantage of using DAGsHub in comparison to MLflow is its capability to keep track of the data. This is achieved by using DVC. Tracking data files with DVC is similar to the use of Git. Files (or even whole directories) have to be added to DVC to be tracked by using the CLI. By doing this, the files are added to the gitignore file and a small .dvc file is created. The .dvc file contains the size of the added file as well as its hash sum. Git versions the .dvc file and the data files can be pushed and pulled to a remote storage, which is better suited for handling large (amounts of) files that might not be text files.

While using DVC in combination with MLflow does not require using DAGsHub, the advantage of using DAGsHub is the unified GUI it provides.

Besides the datasets, also the created models are supposed to be tracked with DVC. However, in comparison to other tools where this can be achieved in the training code, DVC is a CLI tool and thus using it requires more effort in general. In order to facilitate using Git and DVC from the CLI, Fast Data Science (FDS) a wrapper around the two tools has been created by the DAGsHub team [61]. Using FDS combines similar commands of Git and DVC and thus accelerates the usage of both tools.

DAGsHub does not provide any functionality to keep track of the computational environment. However, the basic functionality of MLflow can also be used when using DAGsHub.

The GUI of DAGsHub shown in Figure 8 is familiar to users of the most common Git webservices, but additionally includes a data section, as well as an overview of the experiment runs as known from MLflow, thus most relevant information are combined in one place. It is the advantage of using the open-source tools without DAGsHub. However, most organizations most likely already use a different Git webservice. Especially since DAGsHub lacks functionalities, which other Git webservices provide and, which are often adapted (e.g., CI/CD functionality), organizations might not be willing to migrate to DAGsHub. For this case DAGsHub provides the functionality to mirror another Git repository, which however, contradicts DAGsHub's main advantage of having everything in one place.



☐	TYPE	NAME	TAGS	STATUS	STARTED	UPDATED	ITERATION	PARENT TASK
☐	Training	Keras HP optimization base: General/batch_size=96 Gener...	opt: d2b90906136c4e2fad18b... optimi...	▷ Running	a few seconds ago	a few seconds ago	0	Automatic Hyper-Parameter...
☐	Training	Keras HP optimization base: General/batch_size=96 Gener...	K. e... opt: d2b9090613... opt...	⧖ Pending		a few seconds ago	0	Automatic Hyper-Parameter...
☐	Training	Keras HP optimization base: General/batch_size=128 Gene...	opt: d2b90906136c4e2fad18b... optimi...	✓ Completed	5 minutes ago	a few seconds ago	30	Automatic Hyper-Parameter...
☐	Optimizer	Automatic Hyper-Parameter Optimization	optimization	▷ Running	20 minutes ago	a few seconds ago	8	
☐	Training	Keras HP optimization base: General/batch_size=128 Gene...	opt: d2b90906136c4e2fad18b... optimi...	✓ Completed	9 minutes ago	5 minutes ago	30	Automatic Hyper-Parameter...
☐	Training	Keras HP optimization base: General/batch_size=128 Gene...	opt: d2b90906136c4e2fad18b... optimi...	✓ Completed	14 minutes ago	10 minutes ago	30	Automatic Hyper-Parameter...
☐	Training	Keras HP optimization base: General/batch_size=128 Gene...	opt: d2b90906136c4e2fad18b... optimi...	✓ Completed	19 minutes ago	14 minutes ago	30	Automatic Hyper-Parameter...
☐	Data Pro...	*data_3*	DATASET	✓ Aborted	an hour ago	22 minutes ago	0	
☐	Data Pro...	*data_2*	DATASET	✓ Aborted	an hour ago	22 minutes ago	0	
☐	Training	Keras HP optimization base	Keras example	✓ Completed	38 minutes ago	36 minutes ago	6	
☐	Training	Clone Of pytorch lightning mnist example	MNIST example	✓ Aborted	an hour ago	38 minutes ago	24065	
☐	Training	PyTorch Cifar	PyTorch	Draft		38 minutes ago	0	
☐	Training	TensorFlow mnist example	MNIST	Draft		an hour ago	0	
☐	Training	Pytorch lightning mnist example	MNIST PyTorch example	Draft		an hour ago	0	
☐	Training	LightGBM example		Draft		an hour ago	0	

Fig. 6. ClearML GUI (from [60])

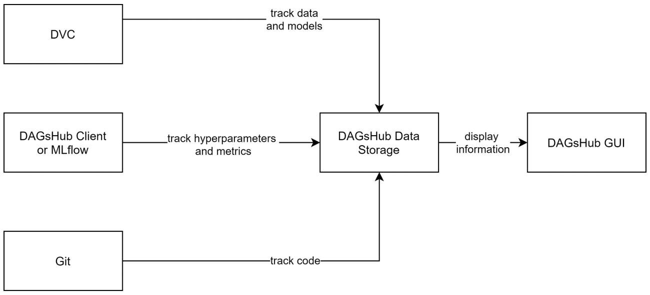


Fig. 7. DAGsHub Architecture

With a free DAGsHub plan, the number of collaborators and storage is limited. Paid plans exist, which allow working in bigger teams. DAGsHub probably has the most potential for teams that already use DVC and/or MLflow and want to keep using the tools but would benefit from unified storage and GUI.

Recent advances of DAGsHub are its integration of Label Studio [62], which can be especially helpful when annotating data in teams. As well as a commenting feature, DAGsHub Discussions, or the possibility to use the GUI familiar from MLflow.

E. Comparison

Table I shows a comparison for most of the defined requirements. As tracking the code is done with Git most of the times and tracking the hyperparameters and metrics and the ease of integration are on a similar level for all four tools, these defined requirements are not included in the table. The tools have different strengths and weaknesses when it comes to ease of use, pricing and more advanced requirements, such as tracking data or computational environment.

MLflow has a well-structured API and can be used for free however, does not provide functionality to track data and automatically keep track of the environment. Also, the effort to set up MLflow in a collaborative environment is more elaborate compared to other tools. Neptune, on the other hand, offers a simple setup and highly functional GUI but requires a paid plan when used as a team and only provides basic functionality to track data and no functionality to track the environment. In comparison to the two previous tools, ClearML handles the tracking of data and the computational environment, taking care of all requirements. Additionally, it is open-source and can be self-hosted or used as a free or paid service. The biggest weakness of ClearML based on our requirements is that because of its richness of features it might not be as easy to use as other tools that might provide less functionality. DAGsHub does not provide functionality to track the environment. However, with DVC the data can be tracked. As a result, DAGsHub can be considered as a good choice for teams already using DVC and MLflow who like to have unified storage and GUI.

V. Conclusion and Future Research Perspective

This paper provided an in-depth analysis of four tools with the focus of tracking machine learning experiments. After describing the process of bringing a machine learning model to production and emphasizing the experimental character of training a machine learning model, requirements for machine learning experiment tools were defined based on the needs of an industrial data science project as well as the research conducted in the field of reproducible machine learning. Additionally 20 tools with functionalities to track experiments,

TABLE I
COMPARITIVE OVERVIEW OF MLFLOW, NEPTUNE, CLEARML AND DAGSHUB

	MLflow	Neptune	ClearML	DAGsHub
Evaluated version	1.29.0	0.16.9	1.7.1	as of September 2022
Data	basic functionality to calculate the hash values and upload it alongside metadata, no way to retrieve the actual data	no dedicated functionality provided	Data Managing and Versioning with ClearML Data	Data Managing and Versioning with DVC
Environment	encourages the user to do it manually (MLflow Projects)	no dedicated functionality provided	automatically keeps track of the installed Python packages and their versions	no dedicated functionality provided
Storing models	easily possible	model has to be stored locally first and can then be uploaded	automatically uploaded if saved locally	possible to store models with DVC, commit required for every upload
GUI	basic GUI	highly customizable & advanced GUI	advanced GUI	unified GUI for data, code, and experiments
Collaboration	possible, requires a shared data storage	possible with a paid account	possible, user limit depends on the operation mode, unlimited for self-hosting	free for public repositories, not free of charge for private repositories
Initial setup and infrastructure	setting up a database or shared file storage is required for collaborative use, alternatively Managed MLflow can be used	easy setup, as the user does not have to take care of the infrastructure necessarily	hosted as well as self-hosting options exist, images to make the setup easier exist	easy setup if DAGsHub is used as Git and DVC storage
Persistence	local file, relational databases, cloud object storage providers	Neptune server, on-premise server	ClearML server, self managed server	DAGsHub Storage, MlFlow backend, cloud object storage providers
Programatic Interfaces	Python, REST-API, R, Java	Python, R	Python, REST-API	Python, REST-API, R, Java (via MLflow)
Workflow support	since 1.27.0 support for pipelines	no native support, but integration into KubeFlow possible	support for pipelines	native support for CI/CD-like pipelines
Ease of Use	intuitive Python API, easy usable by context (with-statement)	intuitive Python API, more explicit method calls necessary, Jupyter notebook integration	intuitive Python API, logs environment automatically	see MLflow

which have been identified in a market research have been presented. Ultimately, four of these tools have been evaluated in detail and compared. This comparison showed all analyzed tools function approximately equally well considering the most basic requirement of tracking hyperparameters and metrics. However, differences exist, considering the more advanced requirements such as keeping track of the data. To conclude, it can be said that the right choice of an experiment tracking tool depends on the specific requirements, and that the open source tool ClearML has been identified as meeting most of the requirements. It has to be noted that due to the quickly changing market of experiment tracking tools, new tools might be released or existing tools might receive new functionality. As a result, further research, also of tools not evaluated in this paper, might be of use.

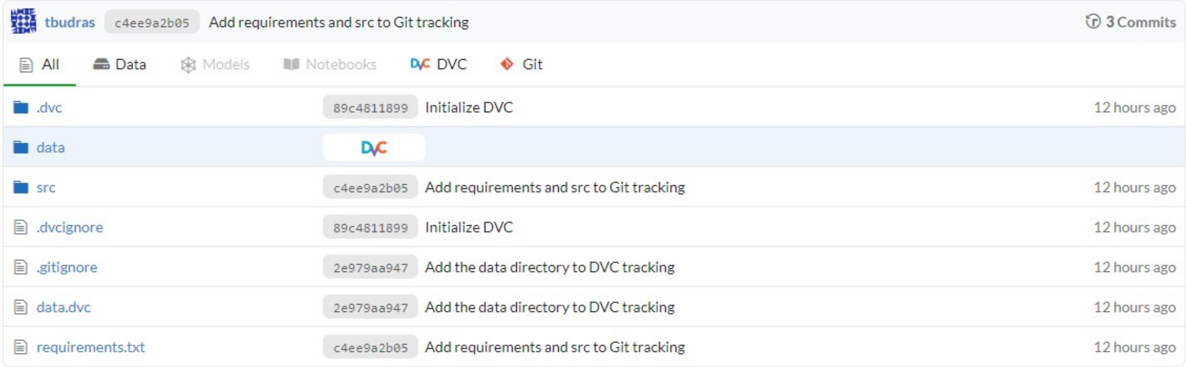
MLflow and Neptune are the tools that have developed recently the most in regard to our requirements. MLflow added a new powerful pipeline feature that eases the training process. Neptune has been adapted to R and now also offers the possibility to self-host a server. Small improvements, e.g., the integration of Label Studio, have been integrated into DAGsHub. In contrast to the other tools, ClearML, which already met most of our requirements, had no major additions. Back then, ClearMl was the favored tool that met our requirements the best. However, especially MLflow and Neptune have caught up.

To better understand why the tools presented in this work have only seen little application in the past, as shown in Sections I and II, further researcher could focus on the difficulties that exist in practice while using such tools. Additionally one or multiple case studies could be conducted, comparing the evaluated tools or a different set using well-defined metrics. Another potential research question could focus on interoperability examining in which cases it makes sense to use multiple tools jointly.

Further, it would be great if standardized formats would be developed to easily switch from one tracking tool to another. This would mitigate platform lock-in effects and improve model life cycle management in the long run. Especially upcoming regulations from administrations [63] that require model documentation and reproducability for longer time periods, would drive such a development.

ACKNOWLEDGMENT

The work is based on the results of a bachelor thesis [64] of the first mentioned author at the company inovex GmbH.



tbudras c4ee9a2b05 Add requirements and src to Git tracking — 3 Commits

.dvc	89c4811899	Initialize DVC	12 hours ago
data	DVC		
src	c4ee9a2b05	Add requirements and src to Git tracking	12 hours ago
.dvcignore	89c4811899	Initialize DVC	12 hours ago
.gitignore	2e979aa947	Add the data directory to DVC tracking	12 hours ago
data.dvc	2e979aa947	Add the data directory to DVC tracking	12 hours ago
requirements.txt	c4ee9a2b05	Add requirements and src to Git tracking	12 hours ago

Fig. 8. DAGsHub GUI

References

[1] T. Budras, M. Blanck, T. Berger, and A. Schmidt, "Comparison of experiment tracking frameworks in machine learning environments," in *Proceedings of the Fourteenth International Conference on Advances in Databases, Knowledge, and Data Applications*, 2022, pp. 21–28.

[2] Machine learning market. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/machine-learning-market-263397704.html (Accessed 2022-12-13).

[3] P. Warden. The machine learning reproducibility crisis. [Online]. Available: https://petewarden.com/2018/03/19/the-machine-learning-reproducibility-crisis/ (Accessed 2022-12-13).

[4] State of data science and machine learning 2020. [Online]. Available: https://www.kaggle.com/kaggle-survey-2020 (Accessed 2022-12-13).

[5] "Ethics Guidelines For Trustworthy AI," EU Commisssion, Tech. Rep., 2019. [Online]. Available: https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60419 (Accessed 2022-12-13).

[6] S. S. Alahmari, D. B. Goldgof, P. R. Mouton, and L. O. Hall, "Challenges for the repeatability of deep learning models," *IEEE Access*, vol. 8, pp. 211 860–211 868, 2020.

[7] P. Nagarajan, G. Warnell, and P. Stone, "Deterministic implementations for reproducibility in deep reinforcement learning," *CoRR*, vol. abs/1809.05676, 2018. [Online]. Available: http://arxiv.org/abs/1809.05676

[8] O. E. Gundersen, S. Shamsaliei, and R. Isdahl, "Do machine learning platforms provide out-of-the-box reproducibility?" *Future Generation Computer Systems*, vol. 126, 07 2021.

[9] M. Mora-Cantallops, S. Sanchez-Alonso, E. Garcia-Barriocanal, and M.-A. Sicilia, "Traceability for trustworthy ai: A review of models and tools," *Big Data and Cognitive Computing*, vol. 5, no. 2, 2021. [Online]. Available: https://www.mdpi.com/2504-2289/5/2/20

[10] R. Garcia, V. Sreekanti, N. Yadwadkar, D. Crankshaw, J. E. Gonzalez, and J. M. Hellerstein, "Context: The missing piece in the machine learning lifecycle," *KDD CMI Workshop*, vol. 114, pp. 32–38, 2018.

[11] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, "Software engineering for machine learning: A case study," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2019, pp. 291–300.

[12] P. Langley, "Machine learning as an experimental science," *Machine Learning*, vol. 3, no. 1, pp. 5–8, 1988. [Online]. Available: https://doi.org/10.1023/A:1022623814640

[13] C. Hill, R. Bellamy, T. Erickson, and M. Burnett, "Trials and tribulations of developers of intelligent systems: A field study," in *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2016, pp. 162–170, ISSN: 1943-6106.

[14] M. Vartak, H. Subramanyam, W.-E. Lee, S. Viswanathan, S. Husnoo, S. Madden, and M. Zaharia, "ModelDB: a system for machine learning model management," in *Proceedings of the Workshop on Human-In-the-Loop Data Analytics - HILDA '16*. ACM Press, 2016, pp. 1–3. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2939502.2939516 (Accessed 2022-12-13).

[15] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[16] N. Hewage and D. Meedeniya, "Machine learning operations: A survey on mlops tool support," *CoRR*, vol. abs/2202.10169, 2022. [Online]. Available: https://arxiv.org/abs/2202.10169 (Accessed 2022-12-13).

[17] "2020 State of Enterprise Machine Learning," Algorithmia, Whitepaper, 2020. [Online]. Available: https://info.algorithmia.com/hubfs/2019/Whitepapers/The-State-of-Enterprise-ML-2020/Algorithmia_2020_State_of_Enterprise_ML.pdf (Accessed 15.12.2022).

[18] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe *et al.*, "Accelerating the machine learning lifecycle with mlflow." *IEEE Data Eng. Bull.*, vol. 41, no. 4, pp. 39–45, 2018.

[19] A. Chen, A. Chow, A. Davidson, A. DCunha, A. Ghodsi, S. A. Hong, A. Konwinski, C. Mewald, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe, A. Singh, F. Xie, M. Zaharia, R. Zang, J. Zheng, and C. Zumar, "Developments in mlflow: A system to accelerate the machine learning lifecycle," in *Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning*, ser. DEEM'20. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3399579.3399867

[20] L. Scotton, "Engineering framework for scalable machine learning operations," Master's thesis, Aalto University. School of Science, 2021. [Online]. Available: http://urn.fi/URN:NBN:fi:aalto-202101311796 (Accessed 2022-12-13).

[21] G. Zárate, R. Miñón, J. Díaz-de Arcaya, and A. I. Torre-Bastida, "K2e: Building mlops environments for governing data and models catalogues while tracking versions," in *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*, 2022, pp. 206–209.

[22] J. Pineau, "Reproducibility, reusability, and robustness in deep reinforcement learning," Paper presented at the meeting of ICLR 2018, 2018. [Online]. Available: https://www.youtube.com/watch?v=Vh4H0gOwdIg (Accessed 2022-12-13).

[23] J. Pineau, "The machine learning reproducibility checklist," 2020. [Online]. Available: https://www.cs.mcgill.ca/∼jpineau/ReproducibilityChecklist.pdf (Accessed 2022-12-13).

[24] R. Tatman, J. VanderPlas, and S. Dane, "A practical taxonomy of reproducibility for machine learning research," 2nd Reproducibility in Machine Learning Workshop at ICML 2018, Stockholm, Sweden., 2018.

[25] Stack Overflow, "Stack overflow developer survey results 2018," 2018. [Online]. Available: https://insights.stackoverflow.com/survey/2018/#work-_-version-control (Accessed 2022-12-13).

[26] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms," Proceedings of the 12th Python in Science Conference in Science Conference (SCIPY 2013).

[27] M. Tabladillo, A. Arora, and C. Gronlund, "What is the Team Data Science Process?" [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/overview (Accessed 2022-12-13).

[28] Aim. [Online]. Available: https://aimstack.io (Accessed 2022-12-13).

[29] Amazon sagemaker. [Online]. Available: https://aws.amazon.com/sagemaker/features/ (Accessed 2022-12-13).

[30] Azure machine learning. [Online]. Available: https://docs.microsoft.com/de-de/azure/machine-learning/how-to-track-monitor-analyze-runs?tabs=python (Accessed 2022-12-13).

[31] Clearml. [Online]. Available: https://clear.ml (Accessed 2022-12-13).

[32] Comet. [Online]. Available: https://www.comet.ml/site/ (Accessed 2022-12-13).

[33] Dagshub. [Online]. Available: https://dagshub.com (Accessed 2022-12-13).

[34] Dominodatalab. [Online]. Available: https://www.dominodatalab.com (Accessed 2022-12-13).

[35] Guild ai. [Online]. Available: https://guild.ai (Accessed 2022-12-13).

[36] H2o mlops. [Online]. Available: https://www.h2o.ai/products/h2o-mlops/ (Accessed 2022-12-13).

[37] Dvc studio. [Online]. Available: https://studio.iterative.ai (Accessed 2022-12-13).

[38] Mlflow. [Online]. Available: https://mlflow.org (Accessed 2022-12-13).

[39] Neptune. [Online]. Available: https://neptune.ai/product (Accessed 2022-13-10).

[40] Paperspace gradient. [Online]. Available: https://gradient.paperspace.com (Accessed 2022-12-13).

[41] Polyaxon. [Online]. Available: https://polyaxon.com (Accessed 2021-07-31).

[42] Tensorboard. [Online]. Available: https://www.tensorflow.org/tensorboard/ (Accessed 2022-12-13).

[43] Valohai. [Online]. Available: https://valohai.com (Accessed 2022-12-13).

[44] Verta. [Online]. Available: https://www.verta.ai (Accessed 2022-12-13).

[45] Vertex ai. [Online]. Available: https://cloud.google.com/vertex-ai (Accessed 2022-12-13).

[46] Weights & biases. [Online]. Available: https://wandb.ai/site (Accessed 2022-12-13).

[47] Data version control - documentation. [Online]. Available: https://dvc.org/doc (Accessed 2022-12-13).

[48] M. Zaharia, A. Chen, A. Davidson, A. Ghodsi, S. A. Hong, A. Konwinski, S. Murching, T. Nykodym, P. Ogilvie, M. Parkhe *et al.*, “Accelerating the machine learning lifecycle with mlflow,” *IEEE Data Eng. Bull.*, vol. 41, no. 4, pp. 39–45, 2018.

[49] Pypi mlflow history. [Online]. Available: https://pypi.org/project/mlflow/#history (Accessed 2022-12-13).

[50] Managed mlflow. [Online]. Available: https://www.databricks.com/product/managed-mlflow (Accessed 2022-12-13).

[51] Mlflow. [Online]. Available: https://www.mlflow.org/docs/1.29.0/pipelines.html (Accessed 2022-12-13).

[52] Mlflow documentation. [Online]. Available: https://www.mlflow.org/docs/latest/index.html (Accessed 2022-12-13).

[53] Pycaret logging with mlflow. [Online]. Available: https://pycaret.gitbook.io/docs/get-started/functions/initialize#experiment-logging (Accessed 2022-12-13).

[54] Mlflow projects. [Online]. Available: https://mlflow.org/docs/latest/projects.html (Accessed 2022-12-13).

[55] Pypi neptune client history. [Online]. Available: https://pypi.org/project/neptune-client/#history (Accessed 2022-12-13).

[56] Neptune r client package. [Online]. Available: https://docs.neptune.ai/integrations/r/ (Accessed 2022-12-13).

[57] Neptune - deploying neptune on your server. [Online]. Available: https://docs.neptune.ai/about/on-prem_intro/ (Accessed 2022-12-15).

[58] Neptune-mlflow integration. [Online]. Available: https://docs-legacy.neptune.ai/integrations/mlflow.html (Accessed 2022-12-13).

[59] T.-W. Huang. tensorboardx. [Online]. Available: https://github.com/lanpa/tensorboardX (Accessed 2022-12-13).

[60] Clearml. [Online]. Available: https://clear.ml/docs/latest/docs/webapp/webapp_exp_table/ (Accessed 2022-12-13).

[61] Fast data science. [Online]. Available: https://github.com/DAGsHub/fds (Accessed 2021-07-20).

[62] M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov, “Label Studio: Data labeling software,” 2020-2022, open source software available from https://github.com/heartexlabs/label-studio. [Online]. Available: https://github.com/heartexlabs/label-studio (Accessed 2022-12-13).

[63] E. Commission. Europe fit for the Digital Age: Commission proposes new rules and actions for excellence and trust in Artificial Intelligence. [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/ip_21_1682 (Accessed 2022-12-13).

[64] T. Budras, “Evaluation of machine learning lifecycle tools in the context of a specific nlp project,” Bachelor’s Thesis, Department of Computer Science and Business Information Systems, University of Applied Sciences Karlsruhe, Germany, 2021. [Online]. Available: https://www.smiffy.de/thesis/thesis-buti1021.pdf (Accessed 2022-12-13).

Automatic Generation of Geographically Accurate Bus Route Maps and its Evaluation

Sogo Mizutani, Yonghwan Kim, and Daisuke Yamamoto
Nagoya Institute of Technology, Nagoya, Japan.
emails: s.mizutani.814@stn.nitech.ac.jp, kim@nitech.ac.jp, daisuke@nitech.ac.jp

Abstract—**There have been many studies on the automatic generation of deformed route maps, but there have only been a few studies on the automatic generation of geographically accurate route maps. This is because mapping route data and bus route data and drawing them on a map are difficult tasks due to various constraints, such as route placement problems. In this study we estimate bus routes that use bus stop coordinate series and strokes and propose an automatic bus route map generation method based on this estimation. In the proposed method, bus stop nodes are first generated on the road network from the bus stop coordinate series. Then, the route between two adjacent bus stop nodes is estimated using the road priority search method, and this is set as the bus route. In the road priority search method, the route with the fewest number of left and right turns between bus stop nodes is estimated as the bus route. Additionally, when drawing multiple routes, the placement order of overlapping sections is dynamically calculated so that intersections when turning left or right are reduced. The experimental results applying the proposed method to 30 bus routes show that the geographically accurate route maps with few intersections among these routes can be generated correctly.**

Keywords-network; bus route map; stroke.

I. INTRODUCTION

The preliminary version of this paper is presented in [1]. This paper includes more detailed evaluations and additional experiments to help understand this work and provides a more detailed discussion about the contribution of this study.

The advancement of public transportation has received considerable attention recently, as typified by Mobility as a Service (MaaS). Among these advancements, buses and bus routes are one of the means of transportation that are at the core of public transportation, and they are of high importance. In a metropolitan area, these transportation systems can be very complex with over hundreds of bus routes. Generally, when using public transportation such as trains and buses, users think about how to get to their destination by looking at route maps. Therefore, there is a need for more understandable and accurate bus route maps.

There are two types of route maps: deformed route maps and geographically accurate route maps. Deformed route maps schematically show the locations and connections of stations and bus stops. As shown in Figure 1, a deformed route map does not necessarily need to be geographically accurate in terms of direction and distance, and knowing the relative positions of stations on a route and their connections is sufficient. Geographically accurate route maps are drawn on a route map based on accurate location information, as shown in Figure 2. As a result, there is an advantage in that it is possible to obtain information about the bus stop and the nearby amenities in addition to its position with respect to the bus line. Moreover, when multiple routes are drawn on one route, the routes overlap each other, thereby reducing visibility. Improving visibility requires arranging routes to minimize overlap between routes. This is called the route placement problem. The route placement problem is a type of combinatorial optimization problem which is known as NP-hard.

Furthermore, online map systems such as Google Maps [2] and OpenStreetMap [3] have become popular in recent years. The online map system allows users to freely change the scale and position of the map and view the desired location. Some APIs, such as Leaflet [4], can control the online map system and permits the drawing of lines and objects on the online map. The use of these technologies enables the expression of geographically accurate route maps by drawing route maps on online maps. Furthermore, the popularization of the General Transit Feed Specification (GTFS) [5] standard has led to transportation system data such as route buses and subways being open to the public. GTFS includes not only timetable data but also bus stop coordinates (expressed by latitude and longitude) and route connection data. Moreover, there has been a problem where the route coordinate series is an optional item and is not necessarily included.

The purpose of this research is to propose a method that generates highly visible and geographically accurate bus routes by estimating routes on road networks from bus stop coordinates and route data included in GTFS and minimizing the overlap between routes.

The remainder of this paper is organized as follows. Section 2 describes the problems that are to be addressed in this study. Section 3 describes the related work. Section 4 outlines the proposed system. The details of the proposed method are described in Section 5. Section 6 reports the experimental results, and Section 7 provides a summary.

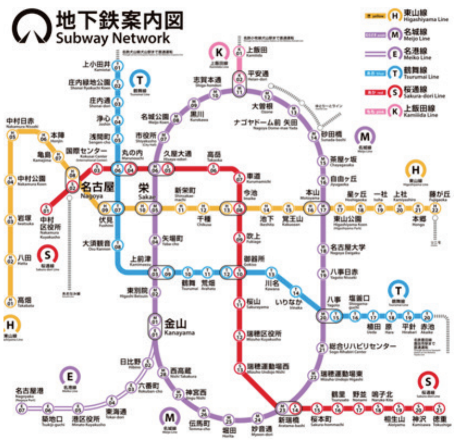


Figure 1. Example of deformed route map (cited from Nagoya City Transportation Bureau.)

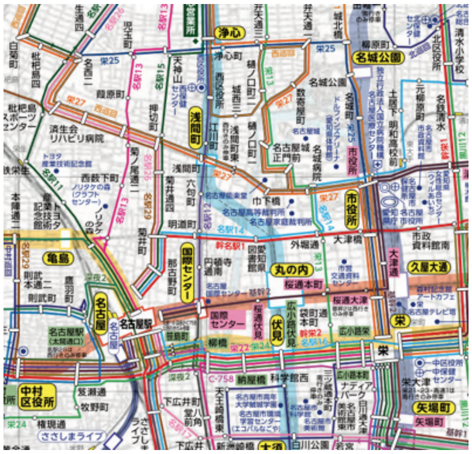


Figure 2. Example of geographically accurate route map (cited from Nagoya City Transportation Bureau.)

II. PROBLEMS

There have been many previous studies on the automatic generation of deformed route maps [6-12], but there have been few studies on the automatic generation of geographically accurate route maps associated with road networks [13]. In practice, most geographically accurate route maps are created manually, but it is very difficult to draw understandable route maps on a road map by associating the road and bus route coordinates. In particular, the number of bus routes is greater than that of railway networks, and there are many routes that overlap each other, so creating these maps require considerable time.

The automatic generation of geographically accurate bus route maps can raise the following issues:

Issue 1) GTFS includes bus stop coordinates, but the coordinates of the routes that connect them are optional items and not necessarily included. Therefore, when trying to draw a geographically accurate route on a road network, the path of the route needs to be estimated from the bus stop coordinates and road network. Additionally, estimating the path requires a bus stop node on the road network. However, the bus stop coordinates indicate the position of the boarding point, and they do not necessarily exist on roads.

Issue 2) A method of determining the placement order between routes by brute force for each road link can be used to minimize the overlap between routes. However, this is an NP-hard problem, and this method has the drawback of exponentially increasing the computational load.

Issue 3) Improving the visibility of the route map requires drawing the routes by shifting them, but the placement order of the routes results in a route map with many intersections. The placement order of routes with fewer intersections needs to be automatically obtained.

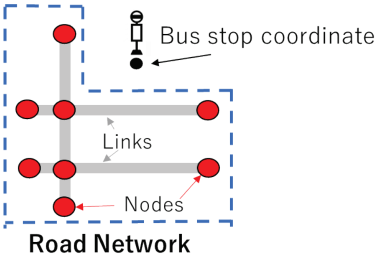


Figure 3. Bus stop coordinates and road network.

Therefore, in this research, we propose a system with the following features:

Feature 1) Stroke-based route generation function

A bus stop node to the nearest point on the road link closest to the bus stop is added from the road network and bus stop coordinates, as shown in Figure 3. Furthermore, the path between the adjacent bus stop nodes is estimated with the road priority search, and that path is set as the bus route path.

Feature 2) Bus stroke (BS)/bus stroke fragment (BSF) generation function

First, a path composed of road links that are estimated by the route path generation function is converted into a BS set composed of strokes. Then, we create a function that generates a BSF set from the BS set that considers bus route overlap. The BS/BSF model can be used to aggregate many road links into the minimum necessary number of BSFs. As a result, the number of combinations can be minimized, and the placement order can be determined efficiently.

Feature 3) Route placement/drawing function

The route placement order is decided from the route map composed of the BSF set to reduce the number of intersections. The routes are drawn on the online map based on these results.

III. Related Work

Previously, research on route maps was mainly research on deformed maps, as shown below, unlike the proposed method. Hong et al. [6] proposed a method for the problem of the automatic generation of deformed route maps of subways. Onda et al. [7] proposed a method for automatically generating railway route maps for Tokyo subway routes. The directions between stations were limited to eight directions (in 45°-increments), and a mixed-integer programming problem (MIP) was used to automatically place route maps while preserving the geographical network topology. There are also many studies on the automatic generation of deformed route maps for subway route maps. Stott et al. [8][10] proposed the automatic generation of railway route maps using a multi-criteria optimization algorithm for appropriate route placement. A clustering method was applied, in which multiple evaluation criteria were set for rendering, and the sum of those results was used as the evaluation value. They proposed a route diagram generation mechanism that thus avoided the local minimum problem and found routes efficiently. Fink et al. [9] proposed a method of drawing routes using Bézier curves for railway route diagrams, which are often drawn linearly. Routes were expressed with the fewest number of Bézier curves using a graph-drawing algorithm based on a dynamic model. Furthermore, Wang and Peng [11] proposed a system that could interactively edit the layout of subway route maps. Route maps are usually drawn with a finite number of colored lines. Lloyd et al. [12] proposed a color-coding method for subway route maps.

An example of research on geographically accurate route maps includes Bast et al. [13] proposed a method of automatically generating geographically accurate route maps by using the connection relationships between stations and route position coordinates included in GTFS as inputs. In this research, they focused on the number of intersections between routes, they improved integer linear programming (ILP) and applied it to the optimization problem of the placement order of routes running in parallel in order to obtain a placement order with few intersections between routes at high speed. Furthermore, as the number of subway routes is small, there was no mention of a stroke reduction method like that of the proposed method, and because subways run underground, there was no need to associate routes with roads, like in bus routes.

In the present study, the concept of a stroke [14][15] is used. A stroke is a grouping of a road network based on cognitive psychology, representing a road that follows a path.

Research using road strokes includes those on road generalization. Zhang et al. [16] achieved road generalization by selecting characteristic roads based on the road connection relationships. Road generalization is a method that draws only major roads in a road network based on the length of the road stroke, and methods that achieve road generalization from facility search results [17][18] and methods that achieve road generalization in a Fisheye view format [19] have been proposed. A path search method using strokes [20] has also been proposed. However, there has been no research that attempts to apply strokes to the drawing of route maps.

IV. Proposed System

In this section, we describe the configuration of the proposed system, data format and terminology definitions.

A. *Configuration of proposed system*

Figure 4 shows the configuration of the proposed system. The proposed system consists of four functions: a stroke-based route path generation function, BS/BSF generation function, route placement function, and route drawing function. The route path generation function generates bus stop nodes from the bus information and road data published in GTFS and generates route paths by searching for routes between adjacent bus stop nodes. The BS/BSF generation function generates a BS by grouping the route data in units of strokes and also generates a BSF by dividing the BS into overlapping sections of multiple BSs. Section 4 describes the details of the definition of BS/BSF. The route placement function sorts based on the rule that sets the placement order of BSF. Finally, the route drawing function draws the route on the online map and presents it to the user.

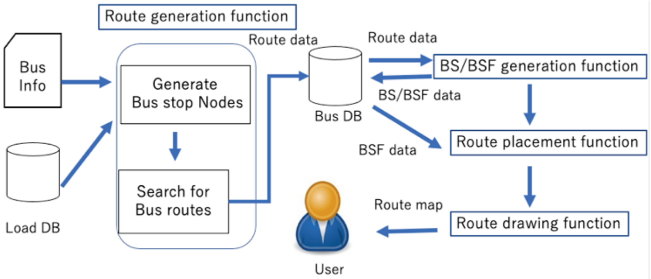


Figure 4. System configuration.

B. *Data format and terminology definitions*

The data formats and terminology used in this study are described as follows:

1) *Definition of road data*

We used OpenStreetMap as a road database. Because “road” is an ambiguous term, this paper defines road data by road links, nodes, and arcs, as shown in Figure 5. Nodes represent intersections and turns on the road network. A link indicates a path that connects nodes. A link has a start node and end node, and it becomes a directed graph given the direction of the road. The shape of the road is represented by a geometry-type arc format that is represented by a point sequence. Table 1 shows the data structure of the road link table. Additionally, OpenStreetMap stores the types of roads, such as highways, national roads, and pedestrian roads, such as road classes. Looped road links where the start node is the same as the end node are not addressed in this study. However, road links with a loop construct can be divided into road links with a non-loop construct by adding a node at the midpoint of the link.

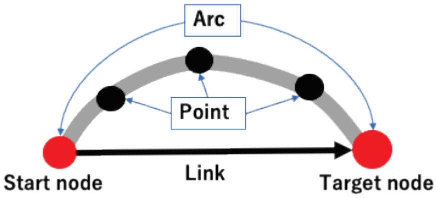


Figure 5. Configuration of road data.

TABLE I. ROAD LINK DATA FORMAT.

Column name	***Data type***	***Explanation***
Id	Integer	Link ID
Clazz	Integer	Road class
Source	Integer	Start node ID
Target	Integer	End node ID
x1	Double	Start node longitude
y1	Double	Start node latitude
x2	Double	End node longitude
y2	Double	End node latitude
Km	Double	Link length
geom_way	Geometry (LineString)	Link shape

2) Definition of stroke

A stroke [14][15] represents a series of road links that follow a path. A road network composed of strokes is called a stroke network. An example of a stroke network is shown in Figure 6 (right). In this example, based on stroke generation rules, the road links on the road network in Figure 6 (left) are grouped to generate a stroke network composed of colored strokes in Figure 6 (right).

Table 2 shows the stroke data format. A stroke consists of an ID indicating the stroke, a series of road link IDs included in the stroke, and its length and shape.

TABLE II. STROKE DATA FORMAT.

Column name	***Data type***	***Explanation***
id	Integer	Stroke ID
link_ids	Text	Included link ID series
stroke_length	Double	Stroke length
arc_series	Geometry (LineString)	Stroke shape

3) Bus data

In this study, we used the GTFS-JP format bus data that was released as open data in 2017 by the Nagoya City Transportation Bureau. We used three items: bus stop data, system data, and bus stop series.

Bus stop data is information that indicates the position and ID of a bus stop. Table 3 shows the data format. Bus stop data includes the bus stop ID, bus stop name, latitude and longitude of the bus stop, etc.

TABLE III. BUS STOP DATA FORMAT.

Column name	***Data type***	***Explanation***
Id	Integer	Bus stop ID
busstop_name	Varchar	Bus stop name
Lat	Double	Bus stop latitude
Lng	Double	Bus stop longitude
noriba_info	Varchar	Additional information
geom_way	Geometry (Point)	Latitude/longitude coordinates

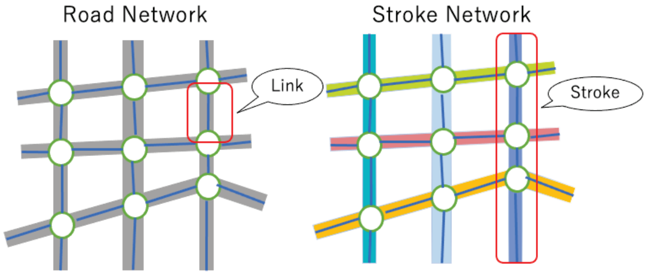


Figure 6. Stroke network.

System data is the data in which operation data is stored for each system of a bus route. The system data is stored in the system table, and information on the start and end points for the operation sections in the system, system code, route code, and direction code are stored. System data is uniquely identified by three items: system code, route code, and direction code.

The bus stop series stores the series of bus stop data that passes from the start point to the endpoint in the operation section of each system data.

V. PROPOSED METHOD

In this section, we describe the details of each proposed method.

A. Stroke-based route path generation function

The route path generation function generates a path on the road network that the bus will actually pass through as a road link series from the bus stop series. The main flow is to generate bus stop nodes from the road network and bus stop series. Stroke-based path search is then conducted on the generated network.

1) Bus stop node generation

As mentioned in Issue 1, a bus stop node needs to be generated from the bus stop coordinates. Specifically, the bus stop coordinates, road link, and road class are set as inputs, and the bus stop node is generated on the road link that is closest to the bus stop in the specified road class.

The bus stop node generation method is shown below. Note that functions starting with ST_ are functions provided by PostGIS [21].

- $L = (l_1, l_2, \cdots, l_i)$: Set of links
- $Class$: Road class
- Bus_{point} : Bus stop coordinates
- Bus_{node} : Bus stop node

Step 1) Among Bus_{point} and L , $Class$ obtains a set of neighboring links other than highways or connecting roads to expressways. Simultaneously, the ST_DWthin is used to obtain a set of neighboring links within 20 m.

Step 2) The link with the shortest distance in the set of nearby links is found using the ST_Distance function and is obtained as the nearest neighbor link.

Step 3) Bus_{point} is used to find the nearest point among the neighboring links, and the ratio r of the nearest point to the start and end points of the link is obtained using the ST_LineLocatePoint function. It is stored in the table as Bus_{node} using the ST_LineLocatePoint function from the obtained ratio r. The data format of the bus stop node is shown in Table 4.

The reason for limiting the road class to those other than expressways in Step 1 is as follows. In urban areas, general roads are often laid under elevated expressways, and it is impossible to determine which bus stop is based on latitude and longitude coordinates alone. Therefore, we used the property that there are almost no route bus stops on expressways and did not generate bus stop nodes on expressways. If the neighboring links are obtained only from the latitude and longitude, then there is a possibility that the wrong links, such as expressways, are obtained.

TABLE IV. BUS STOP NODE TABLE.

Column name	*Data type*	*Explanation*
id	Integer	Bus stop ID
link_id	Integer	Nearest neighbor road link ID
node_lat	Double	Bus stop node latitude
node_lng	Double	Bus stop node longitude
ratio	Double	Ratio on link

2) *Creation of split link*

Nodes on a link require splitting the road links and regenerating the road network to search for a path between bus stop nodes using the created bus stop nodes. In this study, a link that is obtained by splitting a road link at a bus stop node is termed a split link. The procedure for generating a split link is demonstrated as follows:

Step 1) Obtain the nearest neighbor link from the bus stop node table, check how many bus stop nodes there are on the link in the road database, and find the number of splits.

Step 2) If there are multiple bus stop nodes, then they are sorted based on Ratio, which is the bus stop node table ratio, and the links are split, in order, from the starting point.

Step 3) A new ID is allocated to the split link and stored in the split link table.

In the example of Figure 7, Link2, which is the road link of Figure 7 (top), is split at the point of the bus stop node. This increases the number of links from 3 to 4.

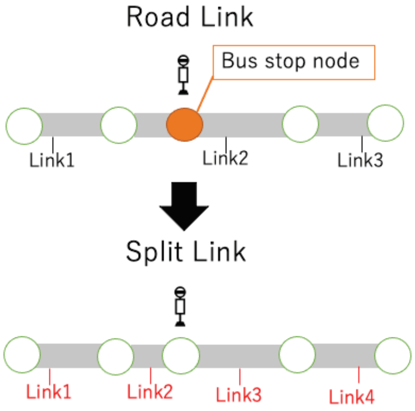


Figure 7. Split link.

3) *Stroke-based route path search function*

The stroke-based route path search function is a method of finding the distance between adjacent bus stop nodes by the stroke-based path search function. The specific steps are as follows:

- ID $= (id_1, id_2, \cdots, id_n)$: Bus stop ID list of the obtained route
- $Node = (N_1, N_2, \cdots, N_n)$: Node ID list
- $V \ni (s, g)$: Combination of start and end points
- $R = (r_1, r_2, \cdots, r_n)$: Path data list

Step 1) Obtain the ID for the specified system, route, and direction code from the bus stop order table.

Step 2) Obtain the bus stop node corresponding to the ID from the bus stop node table and add it to the $Node$.

Step 3) In v $= (N_i, N_{i+1}) \in V$, check if there is a record $(s, g) = (id_i, id_{i+1})$ or (id_{i+1}, id_i) in the path table.

Step 4) If it exists in Step 3, the acquired path data is assumed to be r_i.

Step 5) If it does not exist in Step 3, a path search in v is performed to find r_i.

Step 6) If all paths are found, R is stored in the path table.

In Step 3, the search time can be reduced by checking whether a path is already in the table and finding sections where a path search is not needed.

In Step 5, a road priority search is performed. A road priority search is a method that adopts the path with the shortest distance among paths with the smallest number of passing strokes.

The route bus has a long body, so the costs of turning left or right are high. Therefore, there is a tendency to select paths with as few right and left turns as possible as the bus route. Therefore, it was thought that selecting wide roads as the route path would be better. Thus, we chose road priority search instead of shortest path search, the latter of which is generally used in path search. Table 5 shows the generated path table format.

TABLE V. ROUTE PATH TABLE.

Column name	***Data type***	***Explanation***
route_code	Integer	System code
line_code	Integer	Route code
dir_code	Integer	Direction code
route_name	Varchar	System symbol
s_order	Integer	Start point order number
g_order	Integer	End point order number
s_gid	Integer	Start point bus stop ID
s_name	Varchar	Start point bus stop name
g_gid	Integer	End point bus stop ID
g_name	Varchar	End point bus stop name
geom_way	geometry(LineString)	Path shape
link_ids	Varchar	Link ID series

B. BS/BSF generation function

This subsection describes the definitions and algorithms of BS and BSF.

1) Bus stroke (BS)

BS is a representation of a bus route not as a series of road links but as a series of strokes. Bus routes often pass along roads, so it was thought that expressing a route as a set of strokes instead of as a set of road links could express it with a smaller number of links. Because the number of combinations can be reduced, this is expected to contribute to the speeding up of the combinatorial optimization problem. Figure 8 shows an example of converting a route path (number of links is 6) represented by a series of road links into a BS series (number of links is 3).

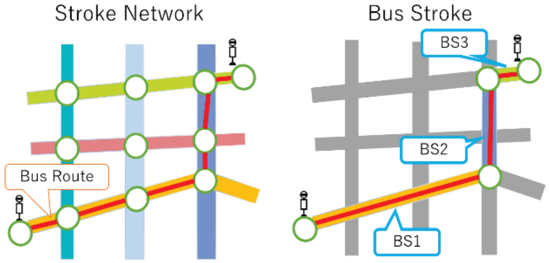


Figure 8. Example of bus stroke generation.

The BS generation procedure is shown as follows:

Step 1) The road link series is obtained from the path data of the desired route.

Step 2) The stroke that contains the acquired link series is obtained from the stroke table.

Step 3) A stroke series is generated in the order of the route paths and stored in the BS table.

The BS table that is generated by the above procedure is shown in Table 6 below.

TABLE VI. BS TABLE.

Column name	***Data type***	***Explanation***
route_code	Integer	System code
line_code	Integer	Route code
dir_code	Integer	Direction code
num	Integer	Order number
stroke_id	Integer	Stroke ID
link_ids	Varchar	Link series
geom_way	geometry(LineString)	Path shape

2) Bus stroke fragment (BSF)

A decomposition of the BS into shorter strokes in consideration of the overlapping of multiple routes is defined as the BSF. Specifically, this is a network in which a path where multiple paths overlap is split at the breakpoints, as shown in Figure 9. In this example, a red route BS1 is split into BSF1 and BSF2 after considering the overlap with the blue route. The BSF generation procedure is shown as follows:

- $R(l_i) = \{r_a, r_b, r_c\}$: Set of routes passing link l_i
- $r.BS = (bs_1, bs_2, \cdots, bs_n)$: BS data list that configures route r
- $r.BSF = (bsf_1, bsf_2, \cdots, bsf_n)$: BSF list that configures route r

Step 1) The $r.BS$ of the desired multiple routes is obtained.

Step 2) A route list $R(l_i)$ that passes through the road links is generated from the bus route data.

Step 3) The BS with multiple routes is divided into overlapping and non-overlapping sections from R, and the overlapping sections are split to generate the BSF.

Step 4) The BSF series $r.BSF$ that passes through each route is obtained and stored in the route BSF table. The BSF table and route BSF table are shown in Tables 7 and 8, respectively, below.

In Step 3, the BS is split based on the obtained R. The BS splitting procedure is shown as follows:

Step 1) $R(l_i)$ and $R(l_{i-1})$ are compared based on the route order. If the included routes are the same, then they are left as they are, and if they are not the same, then l_{i-1} is specified as a splitting position.

Step 2) The BS link series is cut from the first link to the splitting position, thereby splitting the BS.

Step 3) The BSF is generated by splitting the BS and is stored in the BSF table.

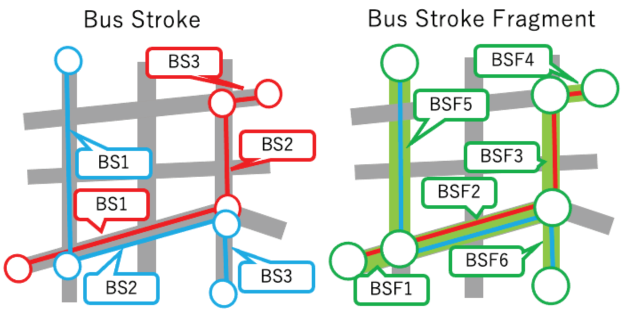


Figure 9. Example of BSF generation.

TABLE VII. BSF TABLE.

Column name	*Data type*	*Explanation*
bsf_id	Integer	BSF ID
stroke_id	Integer	Stroke ID
geom_way	geometry(LineString)	Path shape
link_ids	Varchar	Link series
route_codes	Varchar	Route series

TABLE VIII. ROUTE BSF TABLE.

Column name	*Data type*	*Explanation*
route_code	Integer	System code
line_code	Integer	Route code
dir_code	Integer	Direction code
num	Integer	Order number

Column name	*Data type*	*Explanation*
bsf_id	Integer	BSF ID
bsf_front	Integer	Front BSF ID
bsf_behind	Varchar	Behind BSF ID
geom_way	geometry(LineString)	Path shape

C. Route placement/route drawing function

Details of the route placement function and route drawing function are described below.

1) Route placement function

We describe a method of determining the placement order of parallel sections of multiple routes using BSF as the input for each route. The procedure is shown as follows:

Step 1) The BSF series data for two routes among the input routes is obtained.

Step 2) A target BSF list for which the placement order needs to be determined is created.

Step 3) The placement order of the target BSF is obtained in order from the starting point of the route, and if necessary, the previous placement order is used.

Step 4) One route is added to the current result, and the target BSF list for which the placement order needs to be calculated is obtained, as in the case of the two routes.

Step 5) Steps 3 and 4 are repeated for each input route.

In the above procedure, the following two rules are set to determine the placement order.

Rule 1) The placement order of the target BSF is based on the angle formed by the target BSF and the previous BSF.

Rule 2) For routes where the placement order is not uniquely determined, the BSFs are determined backward until the placement order is determined.

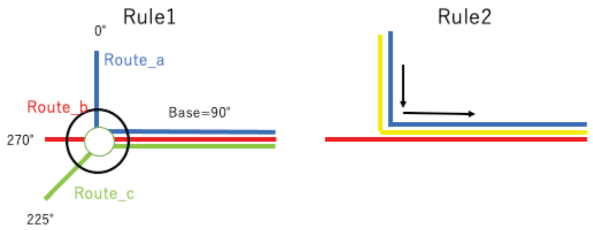


Figure 10. Route placement rules.

Figure 10 shows an example of route placement.

In Rule 1, the order of placement of routes is determined by the angle between the previous BSF and the target BSF at the start point. The PostGIS functions ST_Azimuth and degree were used to calculate the angles: the ST_Azimuth function returns the rightward radians with respect to the north, and the degree function converts radians to degrees.

These are used to calculate the angle between the previous BSF and the target BSF (Base). In the example in Figure 10, (a, b, c, Base) = (0°, 270°, 225°, 90°). The order where these are rotated from the beginning to end until the Base value comes to the beginning, (90°, 0°, 270°, 225°) = (Base, a, b, c), is the placement order arranged in order from the north.

In Rule 2, the angle formed by the BSF connected to the blue and yellow starting points and the target BSF is the same, so the placement order of the front BSF is inherited as is to the placement order of the target BSF.

2) *Route drawing function*

We describe a method that draws a route map whose route placement order is dynamically changed by generating a GeoJSON file as drawing data based on the route placement order results and reading this file.

The generated GeoJSON file is read and drawn onto an online map using Leaflet. At this time, the BSF is drawn using the Leaflet Polyline Offset [22] plug-in, which can give an offset to a polyline to draw routes by shifting them so that the routes in the same section do not overlap. It also has the function of displaying the bus stop position from bus stop data as a marker and displaying the route name in a popup, as well as the function of drawing in one color without giving the route an offset when the scale is small.

Figure 11 shows an example of a Nagoya city route bus drawn using the route drawing function. It can be seen in this example that the three routes and four bus stops (markers) on the map are displayed correctly.

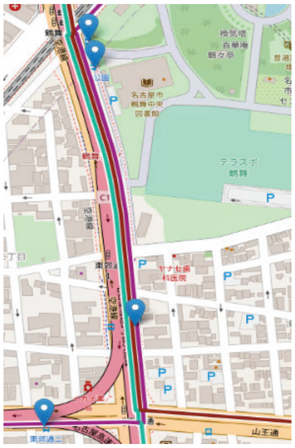


Figure 11. Route drawing example.

VI. Evaluation Experiment

We conducted the following two experiments to verify the effectiveness of the proposed method.

A. *Evaluation of stroke-based route estimation function*

We evaluate the stroke-based route estimation function, which is one of the proposed methods. The route estimation function takes the coordinates of adjacent bus stops as input and estimates the bus route between those bus stops.

As evaluation methods, we compared three methods: road priority search method using strokes (proposed method), shortest path search (conventional method 1), and shortest path search that considers road classes (conventional method 2). Conventional method 2 weights the cost (distance) according to road class when applying the shortest path algorithm.

The evaluation targets were the 50 bus routes of the Nagoya City Transportation Bureau. The evaluation scale was the matching ratio M of the road links between the estimated route and actual route, and equation (1) is used.

$$M = \frac{Number\ of\ matched\ road\ links}{Number\ of\ actual\ road\ links} \times 100 \qquad (1)$$

Table 9 shows the results of the evaluation experiment. The matching ratio M was 92.0% for conventional method 1, whereas the ratios were high at 94.0% and 96.1% for conventional method 2 and the proposed method, respectively. The proposed method was superior to conventional methods 1 and 2 at a significance level of 5%. It was shown that considering the road path, that is, stroke, was effective for bus routes. This was thought to be because the bus tends to run on straight paths as much as possible as turning left or right comes at a high cost.

Figure 12 shows an example of an actual generation. It can be seen that the path generated by the proposed method (Following path) is closer to the actual bus route (Actual line) when compared to conventional method 1 (Shortest Path).

Meanwhile, there were cases where the bus route could not be estimated correctly at some points. In particular, there were many estimation errors near bus terminals. As an example, the green estimated route on the left side was estimated differently on the right side due to an error in estimating the entrance/exit of the bus terminal, as shown in Figure 13. Bus terminals have many bus stops, and the entrances and exits are different, so the road network is also more complicated. Therefore, it is thought that the cause was the generation of a bus stop node on the wrong road link.

TABLE IX. Evaluation of Route Estimation Function.

	Matching ratio M (%)
Conventional method 1	92.0
Conventional method 2	94.0
Proposed method	96.1

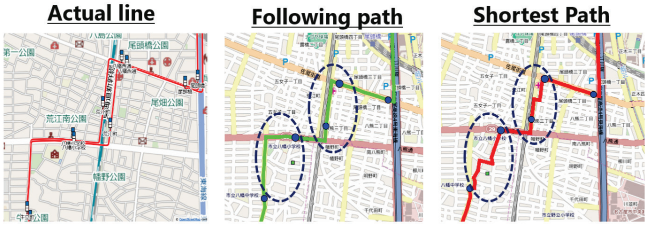


Figure 12. Comparison of route generation methods.

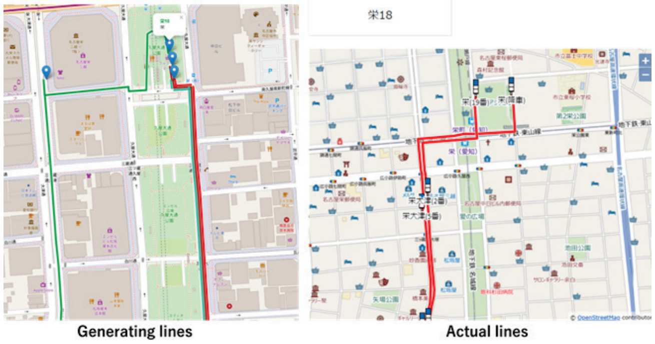


Figure 13. Example of route generation failure.

B. *Verification of link reduction effect by BS/BSF model*

Next, we verified the reduction of the number of links by the BS/BSF model. The BS/BSF model is a method for reducing the number of links by treating bus routes as a set of BSFs rather than as a set of road links. If the number of links can be reduced, then the number of combinations in the route placement problem can be reduced, and the computation time is expected to be reduced.

The evaluation targets were the 661 routes of the Nagoya City route buses. The total number of road links that make up the routes was compared with the number of BSF links that make up the route.

Table 10 shows the experimental results. Each number and reduction rate are shown. The number of road links is 16,433, whereas the number of BSFs is 2,776. As a result, we were able to reduce the number of links by 83.1%.

These results showed that the proposed system can reduce the number of links to minimize the overlap between routes, and that route placement order could be obtained efficiently.

TABLE X. COMPARISON BETWEEN NUMBER OF ROAD LINKS AND NUMBER OF BSFS.

Number of road links	Number of BSFs	Reduction rate (%)
16433	2776	83.1

C. *Qualitative evaluation in automatic generation of 30 routes*

Finally, we conducted a qualitative evaluation on automatically generated routes in this evaluation. The evaluation item here is the visibility of the route map. Figure 14 shows the rendering result of 30 routes. At this scale, 20 routes are displayed on the screen. It was confirmed that the rendering was mostly correct and that there were no problems in actual use.

However, there were several issues. For example, in the route drawing function, routes are distinguished by arbitrary color coding, but visibility is reduced as a result of displaying different routes with similar colors. Additionally, there was the issue of visibility decreasing in drawings of BSFs with three or more routes overlapping or BSFs near bus terminals at positions away from the actual road. Therefore, the realization of a drawing function that maintains visibility near bus terminals and when routes overlap is a topic for future study.

VII. CONCLUSION

In this study, we proposed an automatic estimation method for geographically accurate bus route maps using bus stop coordinate series and strokes. Specifically, the path between adjacent bus stops is estimated using the road

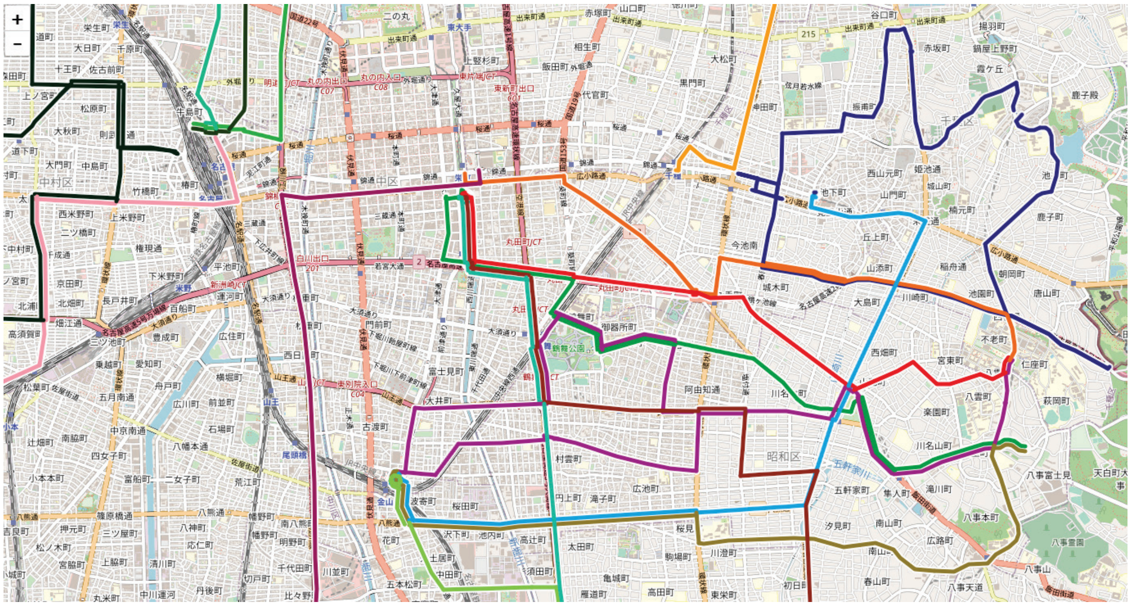


Figure 14. Drawing result of 30 routes

priority search method. The estimated path has the fewest number of right and left turns, so it is thought to be a suitable path for the bus route. As a result, we were able to estimate bus routes with significantly higher accuracy (96.1%) than the conventional method (92.0% and 94.0%).

Additionally, there was the issue where bus routes would intersect each other and become difficult to see when multiple bus routes are drawn on a map. Solving this issue is a type of combinatorial optimization problem, and there is the issue that the computational costs increase exponentially as the number of combinations increases. Therefore, we proposed the BS/BSF model for the purpose of reducing the number of combinations. We minimized the number of links by grouping paths based on the road network to the extent possible. As a result, we were able to reduce the number of links by 83.1% when compared to the conventional road network model.

Furthermore, we confirmed by drawing 30 routes on OpenStreetMap that they could be drawn within a practically acceptable range.

Future issues are as follows. Currently, only 30 routes can be drawn, but we would like to improve this so that it could be applied to more routes. Additionally, we would like to solve the issue of poor visibility at locations where the bus routes intersect in a complicated manner, such as bus terminals. Finally, we would like to investigate the issue of color coding of routes with high visibility.

VIII. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Numbers JP19H04115 and JP21K19766.

IX. REFERENCES

[1] S. Mizutani, Y. Kim, D. Yamamoto, and N. Takahashi, “Automatic generation method for geographically accurate bus route maps from bus stops,” In Proceedings of the GEOProcessing 2022, The Fourteenth International Conference on Advanced Geographic Information Systems, Applications, and Services, Porto, Portugal, ISBN:978-1-61208-983-6, https://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2022_1_50_30037, pp. 19-24, 2022.

[2] Google Maps, (Comment: Example of popular web maps), https://www.google.com/maps/. Accessed 5 Dec. 2022.

[3] OpenStreetMap, (Comment: Data and model used with this research), https://www.openstreetmap.org/. Accessed 5 Dec. 2022.

[4] Leaflet, (Comment: Map library used with this research), http://www.leafletjs.com/. Accessed 5 Dec. 2022.

[5] General Transit Feed Specification, https://gtfs.org/. Accessed 5 Dec. 2022.

[6] S. H. Hong, D. Merrick, and H. A. Do Nascimento, “The metro map layout problem,” In Proceedings of the International Symposium on Graph Drawing 2004, pp. 482-491, Springer, 2004. DOI: /10.1007/978-3-540-31843-9_50.

[7] M. Onda, M. Moriguchi, and K. Imai, “Automatic Drawing for Metro Maps in Tokyo,” IEICE-COMP / IPSJ-AL, 2017-AL163, Vol.13, pp. 1-8, 2017.

[8] J. Stott, P. Rodgers, J. C. Martinez-Ovando, and S. G. Walker, “Automatic metro map layout using multicriteria optimization,” IEEE Transactions on Visualization and Computer Graphics, Vol. 17, No. 1, pp. 101-114, 2010. DOI: 10.1109/TVCG.2010.24.

[9] M. Fink, H. Haverkort, M. Nollenburg, M. Roberts, J. Schuhmann, and A. Wolff, “Drawing Metro Maps Using Bezier Curves,” 20th International Symposium on Graph Drawing, pp. 463-474, 2012.

[10] J. M. Stott and P. Rodgers, “Metro map layout using multicriteria optimization,” In Proceedings of the Eighth International Conference on Information Visualization, pp. 355-362, 2004. DOI: 10.1109/IV.2004.1320168

[11] Y. S. Wang and W. Y Peng, “Interactive metro map editing,” IEEE Transactions on Visualization and Computer Graphics, Vol. 22, No. 2, pp. 1115-1126, 2016. DOI: 10.1109/TVCG.2015.2430290

[12] P. B. Lloyd, P. Rodgers, and M. J. Roberts, “Metro map colour-coding: Effect on usability in route tracing,” In Proceedings of the International conference on theory and application of diagrams, pp. 411-428, Springer, 2018. DOI: 10.1007/978-3-319-91376-6_38

[13] H. Bast, P. Brosi, and S. Storandt, “Efficient Generation of Geographically Accurate Transit Maps,” In Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information, pp. 13-22, 2018. DOI: 10.1145/3337790

[14] R. Thomson and R. Brooks, “Efficient generalization and abstraction of network data using perceptual grouping,” In Proceedings of the 5th International Conference on GeoComputation, pp. 23–25, 2000.

[15] R. Thomson and D. Richardson, “’Good continuation’ principle of perceptual organization applied to the generalization of road networks,” In Proceedings of the 19th International Cartographic Conference, pp. 1215–1223, 1999.

[16] Q. Zhang, “Road network generalization based on connection analysis,” In Proceedings of the 11th International Symposium on Spatial Data Handling, pp. 343–353, 2005. DOI: 10.1007/3-540-26772-7_26

[17] D. Yamamoto, M. Murase, and N. Takahashi, “On-Demand Generalization of Road Networks based on Facility Search Results,” IEICE Transactions on Information and System, Vol. E102-D, No. 1, pp. 99-103, 2019. DOI: 10.1587/transinf.2017EDP7405

[18] M. Murase, D. Yamamoto, and N. Takahashi, “On-demand Generalization of Guide Maps with Road Networks and Category-based Web Search, Results,” In Proceedings of the 14th International Symposium on Web and Wireless Geographical Information Systems, Vol. 19, pp. 53-70, 2015. DOI: 10.1007/978-3-319-18251-3_4

[19] D. Yamamoto, S. Ozeki, and N. Takahashi, “Focus+Glue+Context: An Improved Fisheye Approach for Web Map Services,” In Proceedings of the ACM SIGSPATIAL GIS 2009, pp. 101-110, 2009. DOI: 10.1145/1653771.1653788

[20] Y. Hiura, (supervisor: D. Yamamoto), “Proposal of an efficient nth min stroke shortest path search method,” Master’s thesis, Nagoya Institute of Technology, 2020. (In Japanese)

[21] PostGIS, (Comment: Database used with this research), https://postgis.net. Accessed 5 Dec. 2022.

[22] Leaflet Polyline Offset, (Comment: Map library used with this research), https://github.com/bbecquet/Leaflet.PolylineOffset. Accessed 5 Dec. 2022.

State of Affair in Terms of Big Data Utilization in Complex System Engineering Organizations: A Case Study in the Context of Norwegian Industry

Fahim A. Salim
Dept. of Science and Industry Systems
University of South-Eastern Norway
email: fahim.a.salim@usn.no

Haytham B. Ali
Dept. of Science and Industry Systems
University of South-Eastern Norway
email: haytham.ali@usn.no

Tommy Langen
Dept. of Science and Industry Systems
University of South-Eastern Norway
email: tommy.langen@usn.no

Pragna Labony Deb
Dept. of Science and Industry Systems
University of South-Eastern Norway
email: pragnalabony.sust@gmail.com

Andreas Wettre
Centre for Design Research
Oslo School of Architecture and Design
email: andreas.wettre@aho.no

Gerrit Muller
Dept. of Science and Industry Systems
University of South-Eastern Norway
email: gerrit.muller@gmail.com

Kristin Falk
Dept. of Science and Industry Systems
University of South-Eastern Norway
email: kristin.falk@usn.no

***Abstract*—Effective utilization of big data is still an open question for most organizations. In the presented case study, we attempted to get a nuanced understanding of the state of affairs regarding big data utilization in Norwegian high-tech industries. This case study uses research methods like questionnaires, semi-structured interviews, observations, and co-creation sessions. These methods explore the data utilization processes at partner organizations of the H-SEIF2 consortium or lack thereof to systematically utilize big data in their projects from the perspective of employee perception. The presented case study provided insights into the case study organizations. For example, organizations still heavily rely on inconsistent manual data logging and our survey found that the Project Managers have a more optimistic perception of their usage of big data. In contrast, upper management has a more modest opinion of their current state. The presented case study also provided a more in-depth analysis of challenges that hinders data utilization and identified opportunities to enhance the value of ongoing and potential digitalization initiatives at the organizations.**

Index Terms*—*Questionnaire; Big data; Early Phase Decisions.

I. Introduction

The paper is an extended version of the article presented at the Modern Systems Conference 2022, aiming to get a nuanced understanding of big data usage in the context of Norwegian Industry [1].

Big data analytics (BDA) and digitalization is a trending topic and is expected to be a big asset and an engine for innovation that has the potential to propel new technological revolutions [2]. The benefits of using big data in enhancing operations in general and in project life cycle management are well understood by organizations of all types and sizes [3]–[6]. However, how to do so effectively is still an open questions for most organizations [4] [7] [8]. For example, a study by Qlik and Accenture states that over 74% of employees feel anxiety with when working with data [9]. Similarly, a recent study by Rackspace Technology [3] reported that organizations perceived a rise in difficulty in terms of utilizing Big Data Analytics (BDA) and specifically Artificial Intelligence (AI) and Machine Learning (ML).

Researchers have identified multiple challenges that limit organizations' ability to enhance big data utilization. These challenges include a lack of common language among engineers, ineffective knowledge sharing, and difficulty in finding system information, to name of few [10]–[13]. The presented paper focused on the users of big data, specifically the internal users such as employees working on projects.

The presented case study in the context of the Norwegian high-tech industry is part of the H-SEIF2 [14] project. The case study and the analyses are being performed in close collaboration with the H-SEIF2 consortium industry partners to provide a nuanced understanding of the state of affairs at Norwegian high-tech companies in terms of big data utilization. The case study is designed using a combination of techniques such as industry as a laboratory [15], Co-creation [16], questionnaires and semi-structured interviews [17]–[19].

The remainder of the paper is structured as follows. The following section (section II) describes the related work. We describe the design of the case study in section III. This is followed by results from analysis and observations (section

This research is part of the H-SEIF 2 project and is funded by the Research Council of Norway through the innovation project (IPN project no.: 317862).

IV) and concluding remarks (section V).

II. Related Work

Related work describes the notion of big data as described in systems engineering. It further discusses related surveys conducted by other researchers.

A. Big Data and Complex System Engineering

Regarding the notion of big data utilization in complex systems environments, there are many definitions for data, also called big data. Many authors and practitioners have defined big data as the notion of (Vs). Some authors [20]–[22] have defined big data in terms of the 3Vs: Volume, Velocity, and Variety. Others [23]–[25] have extended the definition by adding value as the fourth V (4Vs).

In addition, M. White [26] suggested adding Veracity as the fifth V (5Vs). In this context, "Volume" refers to the vast amount of data. "Velocity" refers to the speed at which new data is generated, whereas "Variety" represents different data types. The fourth V, "Value," refers to how we can benefit from big data by turning it into value. "Veracity" includes biases in the data and strives to encompass the level of sufficiency or insufficiency of the data [27].

The elements in our world are connected and dependent on each other. The complexity is higher than ever and is continuing to be more complex. Products are linked in an intertwined network of dependencies, and services rapidly develop to satisfy demanding customers. Fig. 1 shows an example of an intertwined network of dependencies.

It visualizes an elaboration of the characteristics of what Schätzet et al. call CPS (Cyber-Physical Systems) [28]. CPS is closely related to several concepts such as (Big) data and its analysis, the Internet of Things (IoT), Systems of Systems (SoS), Mechatronics, and Embedded Systems. We also add the sociotechnical aspects within the other three circles: Human Social Organizational, Innovation Ecosystems, and Political aspect.

Systems are also adapting in a dynamic behavior to technical and social factors. Thus, there is a need for companies to explore new ways to maintain competitiveness. It is crucial for these companies to use the most effective methods and that these are sufficiently founded. A solution is to utilize data early in product and service development.

Organizations and their employees recognize that big data analytics will be a significant source of competitive advantage in the future. Still, several impediments inhibit them from fully utilizing big data's benefits. Most technologies were not developed to satisfy the expanding demands of big data analytics [29]. Employees who want to exploit the power of big data may run into significant issues due to data complexity and inherent messiness. Moreover, digital data is often stored in various forms, such as unstructured databases and discrete different text files [30].

Lack of data analytics skills among current employees may increase data entry mistakes, resulting in misinterpretation and loss of important information and ultimately reducing the value of the data [31]. Ethical considerations like privacy and cultural barriers are also hurdles regarding big data usage. For example, some businesses know how big data might help them improve their operations. Still, cultural or technological limitations prevent employees from using big data in production.

B. Survey Questionnaires Regarding Big Data Utilization

Questionnaires and surveys are widely used for different purposes, e.g., comparing two products or services or both [8] [32]. They are often used to understand the needs of perspective users of future products and services, or as part of user-centric design [19] [17], or to collect data on customer, employee, or student satisfaction, [33]–[35].

Regarding the utilization of big data in the operations of organizations, Qlik and Accenture [9] conducted a survey to understand big data utilization in enterprises. They found that 60 to 70% of the collected data in an enterprise is never used and a vast majority of about 74% of employees feel overwhelmed or simply unhappy working with data. They also found that only 37% of employees trust their decisions more when they are based on data, while 48% preferred gut feeling over data-driven decision-making [9].

Focusing on high-level decision makers, a survey by Rackspace Technology [3] found that employees perceived difficulty with ML and big data has been increasing. The survey [3] also found that employees considered data dispersed across many different systems to be one of the most significant barriers to drawing insights from it. Lack of skillset and talented employees is perceived as a considerable concern and limitation in fully utilizing big data in enterprise decision making [3] [9] [36].

Raguseo [7] focused on the CIOs of French medium and large enterprises to understand differences across industries and the size of organizations. Analysis of the questionnaire found that the organization's size influences investments in technologies like ML software tools. The author did not find any statistical differences across different industries.

While employees often appear in the discourse and are considered a crucial element in utilizing big data systems, they are often the ones most neglected [37]. Moreover, most of the research mentioned above focused on high-level decision makers, not a cross-section of employees, departments, and job roles.

The presented paper is part of the H-SEIF2 [14] research project that aims to develop a human-centered framework for utilizing big data during early phase decision-making in the product development process. By focusing on Norwegian industry partners, the presented work extends the related work by focusing on a cross-section of employees to understand better the differences among different departments, employee profiles, and across various industries.

III. Case Study Design

The primary goal of the case study is to understand the current state of affairs in terms of big data utilization at the partner organizations in the H-SEIF2 [14] project. The H-SEIF

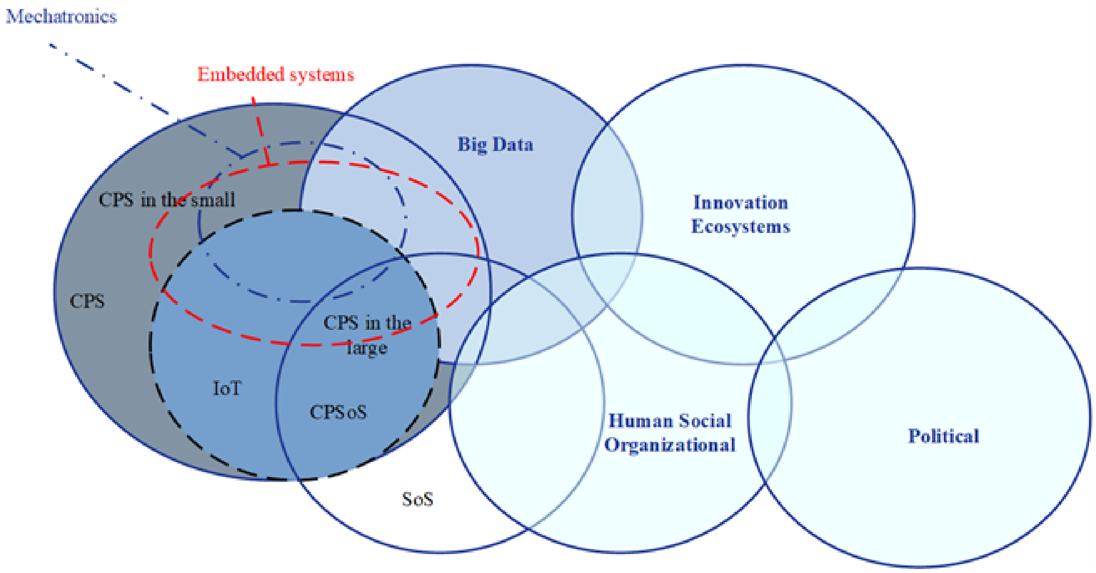


Fig. 1. Complex Interlinks

2 project aims to harvest the value of big- data to enhance the experience of stakeholders during complex system engineering projects by collaborating with industry partners to improve their digitalization efforts. The goal is to design data-driven frameworks and methodologies to allow the industry partners in data-supported early-phase product development decisions.

The presented case study, following the aims of the H-SEIF2 project of harvesting the value of big data for industry partners; evaluates the question: *How are the different industry partners utilizing big data in their operations*?

1) What are the gaps, opportunities, and barriers to enhancing the utilization?
2) Are there any differences in different organizations and departments within an organization, and what can the partners learn from each other's experiences?
3) What can partner organizations do to maximize the value of ongoing or potential digitization initiatives?

A. Methodology

We used a combination of workshops, interviews, surveys, on-site observations, and subject-matter expert feedback from the industry and academia to design the questions for the survey. An adapted version of the Applied Research Framework was used as the research method to design this survey [38], [39]. The framework consists of the following steps:

Step 1: Shape the line-of-reasoning. In this step, the line of reasoning is expressed by following the structure of *Problem-goal-solution-rationale*. Additionally, the research questions we formulated more specifically based on the broad problem statement we expressed within the line of reasoning. The main research question was establishing a baseline for the H-SEIF 2 research project consortium.

Step 2: Explore literature. In this step related studies from literature to aid in designing the questionnaire (see section II-B for details).

Step 3: Elicit expert opinions. Expert's views, in this context, refer to the domain expert among the scholars within the research methods. Semi-formal discussions with different experts were conducted. For example, two of the co-authors have decades of consultancy experience. Several workshops were conducted with experts from academia and industry to gather their feedback regarding good, best, and emergent practices regarding the survey's design and questions.

Step 4: Determine the research design. Notes were kept from the workshops and related literature using a shared platform. Furthermore, steps 2 and 3 were performed iteratively.

A total of 6 companies participated in the presented study.

In the first stage, a survey was conducted with a cross-section of employees at different industry partners. A total of 5 companies from the H-SEIF2 consortium participated in the survey. This included a technology consultancy, an autonomous transportation solutions provider, an oil and gas company, a Data Agency and an Industrial conglomerate (3 divisions/subsidiaries). Further, we conducted semi-structured interviews at the technology consultancy and co-creation and observation sessions with different partners (Automated Parking System (APS), the oil and gas company, and an industrial conglomerate's defense division) to better understand the state of affairs at the individual organization.

The partner organizations vary in size and they work in different sectors; in the second phase, we had more direct interactions with some of the individual partners to better understand the state of affairs at the company. We customized the approach per partner organization based on factors such as the size and time commitment of the organization, the type and duration of projects organizations are involved in, the industry sector to which the organization belongs to and the level of access granted to researchers by each organization. For example, some co-authors work closely with partner organizations, allowing them more opportunities to observe and hold co-creation sessions. While at some partners, semi-structured interviews were conducted as co-creation sessions were not feasible due to the limited availability of the involved personnel.

B. Survey Questionnaire

Based on the methodology described above (section III-A) the finalized version of the survey questionnaire consisted of 24 questions in 5 categories. We incrementally developed the survey's language, style, and structure with continuous feedback from participating subject-matter experts and practitioners from industry and academia.

We wanted to cover many aspects in our survey. We formed questions about the data **availability**; if the data is challenging to find, it is also difficult to utilize. This category also goes to data ownership, as external organizations own the data, and there might be obstacles to utilize.

Then if you have the data, the data needs to be **usable**, contextualized so that it is possible to transform to value and use the analyses in the decision-making process focusing on early phase product development.

We wanted to know about data integrity. Suppose users do not **trust** the data, thinking it is extracted from sources or through a process that reduces the reliability or presented in ways that make you doubt what the data is saying. In that case, it will most likely not be utilized.

Then even if the datasets are reasonable, there might be **processes, politics, habits, time** or a lack of competence that prevents the organization from using the data for decision making. Sometimes, the essential decisions are made on gut feelings, emotions or based on older experiences from similar projects.

We asked the participants to rate the extent to which they agreed or disagreed with the statements on a 5-point Likert scale. In the initial phase, we collected 40 responses from employees at partner organizations.

C. Semi-structure Interviews

The survey was followed by semi-structured interviews with employees working in the technology consultancy. The consultancy works on different project types with variable duration and activities they performed on those projects. For this research, we focused on the personnel working on one project the consultancy did for one of its clients.

The interviews focused on how the consultancy acquired insight about how they utilized big data in their previous and current projects considering the project team as an example. The in-depth interviews allowed us to achieve a broader understanding of the point of view of the engineers and managers at the consultancy, which facilitated a qualitative analysis [40]. We designed the semi-structured interviews for eliciting information about specific topics [41], [42]. The interview guide consisted of 20 main open-ended questions that reflected the stakeholder needs while using big data. Fig. 9 shows the interview guide.

We conducted all the interviews using Microsoft Teams, a teleconference tool widely used for video conferences and taking interviews. We recorded each interview using the built-in mobile recording feature, and transcripts of these recordings were generated by Office Dictation, powered by Microsoft speech services, and embedded in Microsoft Word. We analyzed the data later by following thematic analysis, a commonly used approach for qualitative studies.

The details about the thematic analysis and its outcomes are described in section IV-B.

D. Co-creation and Observations

We conducted multiple co-creation and observation sessions with industry partners at the Automated Parking System (APS), the oil and gas company, and an industrial conglomerate's defense division.

The APS provider is a medium-sized company. It delivers APSs and provides maintenance services in operation. The company has around 35 parking installations throughout Norway.

The energy services (oil and gas) company is a multinational corporation that provides life cycle services for the energy industry. We conducted multiple workshops and observation sessions to understand the real-life context of the company involved in complex engineering projects.

The industrial conglomerate have divisions in areas such as Shipping, Defence, and financing, to name a few. We focused on the defense and aerospace division's employees for the presented paper.

IV. Results and Observations

This section details results and observations from the survey analysis.

A. Analysis of survey responses

The questionnaire results (see Figs. 2, 3, 4, 5 and 6) show that internal stakeholders (employees) feel dissatisfied by the utilization of big data in their projects, especially in early phase decision making as the Net Promoter Score (NPS) is negative across the board.

Regarding data availability, the respondents either agree or are partial to the availability of data they need (see Fig. 2 Q1 to Q3) although they do think there is room for improvement as the NPS is negative. However, the availability of the right tools to explore and process the data is a more significant issue for them; as they mostly disagree or have a neutral response to the questions regarding the availability of such tools (see Fig. 2 Q4 to 6). One notable surprise, however, is question# 7 (see 2), which asks the respondents if data is being held back from them for confidentiality reasons, to which the respondents disagreed. In this case, it is a positive outcome and runs counter to our earlier assumptions [43].

The respondents expressed more dissatisfaction with the usability of data compared to its availability (see Fig. 3). For example, respondents largely disagree with whether they spend sufficient time analyzing past data in the beginning

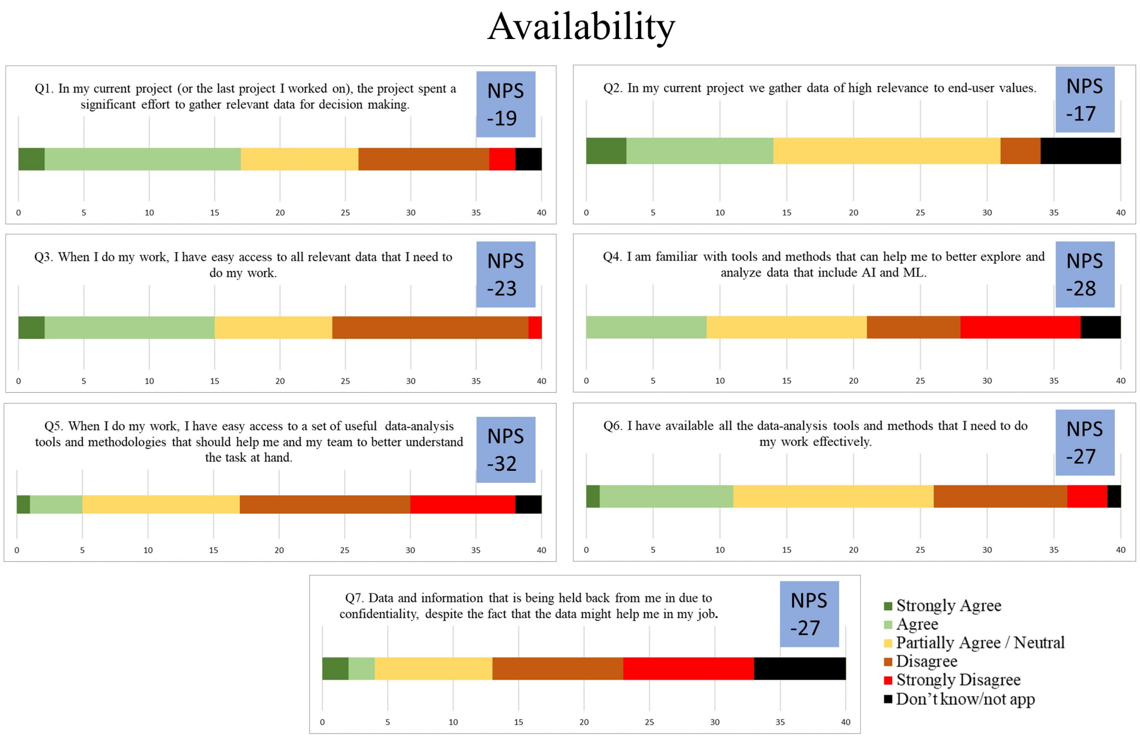


Fig. 2. Responses to Questions about Data Availability. NPS stands for Net Promoter Score

phases of their projects (Q# 8). They also think the procedures for sharing data at their organizations are insufficient (Q# 13). The respondents also have negative or neutral sentiments regarding utilizing past (historical) data as lessons learned for new projects (Q#14).

In cases when data is available to them, the respondents expressed trust in the integrity and correctness as the responses to Q# 15 are positive or neutral (see Fig. 4).

The respondents are somewhat divided on the competence of using (big) data. Respondents avoided this category of questions more often than other questions, and responses have a relatively even split (see Fig. 5).

Respondents believe their organizations have a long way to go when utilizing big data in their operations and project development. Respondents mostly disagreed with the question related to organization behavior (see Fig. 6).

For the most part, the survey results are not surprising, as it is not only the finding of our initial conversations with industry partners, but other surveys reached the same conclusion [3], [9]. However, there are some interesting findings as well.

For example, one interesting outcome is that engineers and personnel involved with the technical aspect of projects gave lower scores than project managers and upper management. Project managers seem to have a rosier perception than others (see Fig. 7). There is a need for more outstanding communication among project managers and other non-technical stakeholders, engineers, and technical personnel. While the more positive responses are somewhat in line with [9], there are notable differences compared to [9]. For example, in our survey, the upper management seemed less optimistic than the project managers.

Another notable exception is the "Competency" section of the questionnaire. For example, the report [9] stated that business leaders overestimate the capabilities of their workforce. In contrast, our survey showed that engineers and project managers gave higher responses than upper management (see Fig. 7).

In terms of age groups, employees in younger and older age groups overall gave higher scores compared to the middle (35-44) age group, while the middle age group reported the most confidence in their competency compared to the others (see Fig. 8). Also, regarding the organization behavior category, younger and senior employees express greater optimism than the 35-44 age group. Question# 21 was an exception, which asks about taking full advantage of operational data in early phase decision making, to which the 35-44 age group gave a higher score than the others.

B. *Thematic analysis of semi-structured interviews*

To extract themes from the interviews, we used a technique known as thematic analysis [44] for finding, analyzing, organizing, summarizing, and reporting the outcomes from the data collected [45]–[47]. Fig. 9 outlines the utilized approach.

Interviews transcripts comprised of 26 pages, and we added them as input for the NVivo, a qualitative data analysis software. In the first step, we read the transcripts to get a

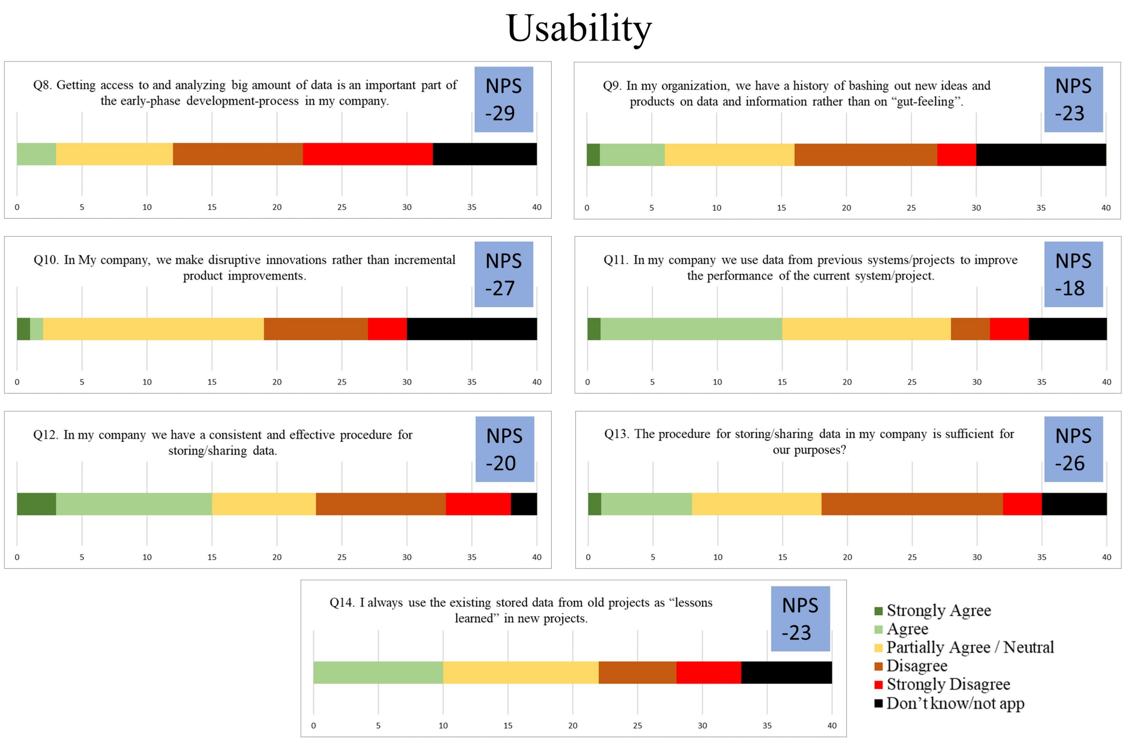


Fig. 3. Responses to Questions about Data Usability. NPS stands for Net Promoter Score

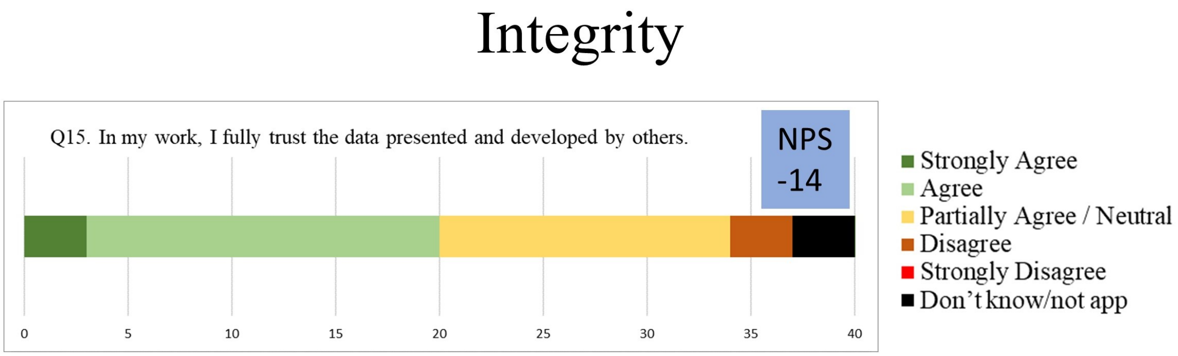


Fig. 4. Responses to Questions about Data Integrity. NPS stands for Net Promoter Score

broad picture of the collected data and familiarize ourselves with the text. Important phrases and words were identified that were repeatedly used by participants and assigned codes. Then, in order to facilitate analysis, the codes were grouped together into larger categories based on their shared qualities such as personal beliefs or professional progress [48]. In the second stage, we used NVivo software to create a hierarchical category system based on the linkages or ties between codes and categories. We searched and identified patterns across the data. These patterns were considered themes. We identified and iterated several times over potential themes that we identified from the codes and categories. These iterations ensure that we included all relevant data from the interviews.

In qualitative research a code refers to a word or phrase that captures the meaning or essence of a piece of data. Codes can be organized in to categories to further get a nuance understanding of the data in this case the interview transcripts. Using a tool such as NVivo, certain themes can be extracted from that. In NVivo, a theme is a topic that is found within the data.

We spotted the crucial statements based on the themes, including codes and categories with descriptions, using the

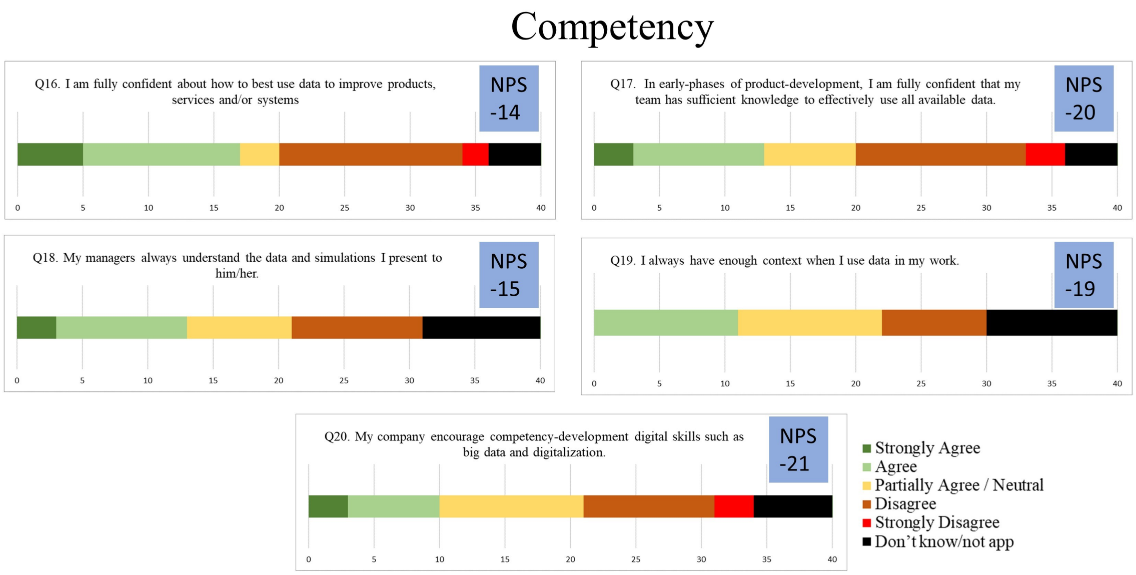


Fig. 5. Responses to Questions about Competency. NPS stands for Net Promoter Score

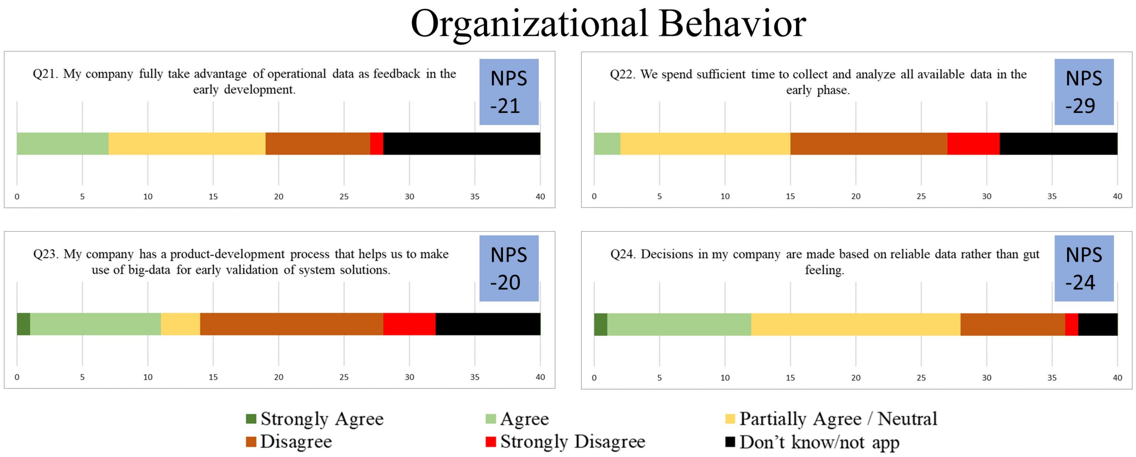


Fig. 6. Responses to Questions about Organization Behavior. NPS stands for Net Promoter Score

NVivo software tool. Furthermore, we calculated the frequency of the categories and sub-categories. Ultimately, we visualized the results to highlight the most frequently referred categories that emerged from the interview data.

We identified 11 codes in four categories. These categories (and codes) further belong to three themes. Fig. 10 visualizes the generated code-book with descriptions. Fig. 12 depicts the codes, categories, and themes and their relation.

Calculation of relative frequencies of the identified codes and categories demonstrated that the most cited categories are data handling in projects (49.1%), data types and tools (17.6%), and professional development (17.6%). Focusing solely on the codes, we can see that access to past data is the most cited, with 18.5%, followed by data storage (14.8%), presence of analysis tools (11.1%), detect certain data (10.2%); those four sub-categories totaled 54.6% of all occurrences. The other 7 subcategories total 45.4% (Fig. 11).

We identified three main themes from the categories (Fig. 12):

- Reflection on big data usage.
- Utilization of big data in projects.
- Approaches to establishing the data-driven culture.

Theme 1: Reflection on Big Data is a significant theme that reflects the experiences and feelings of the employees on big data usage at work. The theme portrays the whole story of the stakeholders' viewpoints on this project and is depicted in

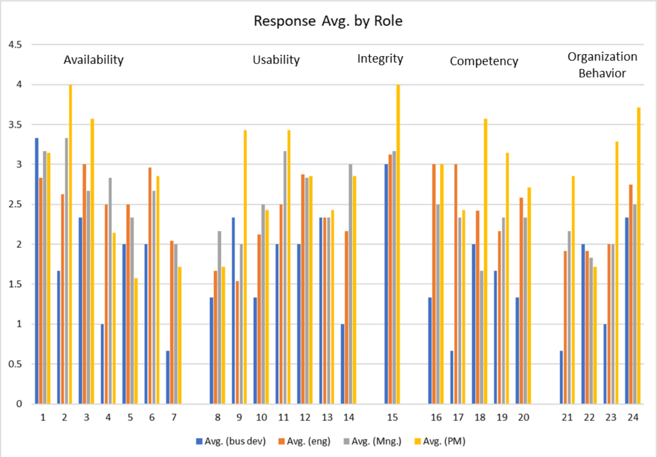


Fig. 7. Average responses per job roles.

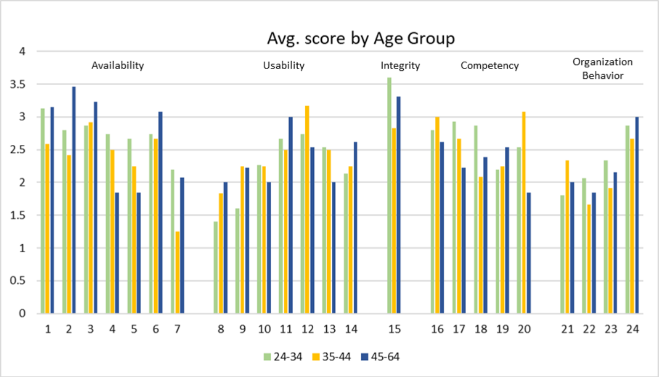


Fig. 8. Average responses per age group.

codes and categories. The most common words in this category were important, useful, better, and right (Fig. 13).

Theme 2: Utilization of Big Data in Projects depicts how the employees use big data and what issues they face in the organization. Codes and categories also illustrate data tools and the data usability process. For instance, when we asked about access to the relevant data from past projects, the respondents revealed they don't have access to past (historical) data due to a lack of a central storage system and privacy. In the final analysis, we used this theme to identify employees' current main issues. However, one respondent noted it differently:

"*The case is also that we don't have some that much data from the other projects. We have not been good enough at collecting data and storing the data, so even if it were relevant, we wouldn't have access to a lot of it. But if I want to answer the question, it could be relevant if we spend more time analyzing and understanding how we can use data between projects.*"

The replies from the employees suggested that the consultancy lacks essential data access and storage facilities and the ability to comprehend and devote time to data analysis.

Theme 3: Approaches to Establish the Data-Driven Culture conveys the approaches the organization is taking to create a data-driven environment and the competency level of the company's employees. Employees feel the need to develop their skills in data analysis more, although the management appears to be interested in such affairs of the organization. We know from in-depth interviews that the users have knowledge about data analysis, but the organization also need to emphasize more workshop or courses regarding on big data or digitalization skills to improve their data literacy.

C. Observations from co-creation sessions

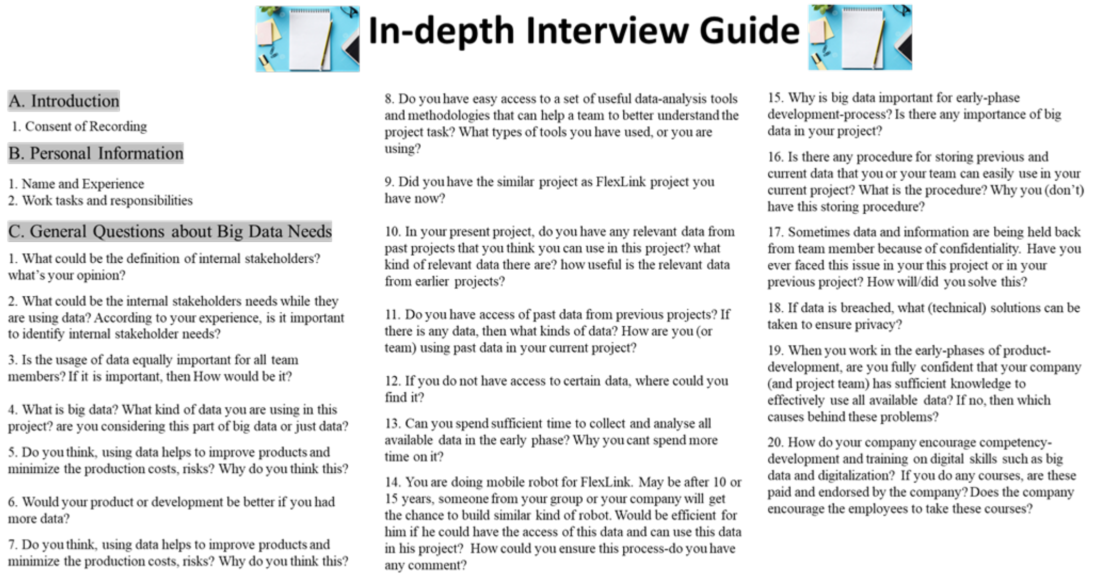
This section details the observations from co-creation sessions at the APS provider, the Oil and Gas company and an Industrial Conglomerate.

1) Observations at the Automated Parking Systems (APSs) Provider: The co-creation and observation sessions revealed that the APS provider has mostly unstructured data with many variations. One primary source of this data is maintenance log data, also called failure data. Maintenance personnel are logging failure data manually using an Excel file. Failure data includes a description of the failure events, its possible cause, and a possible implemented solution to the failure called the reason parameter (column) in the company's failure data. In addition, the failure data include, among others, the following parameters (columns): date (for a maintenance/failure event), time, telephone number (for the maintenance personnel who investigated the failure event), place number (for which parking lot the failure event occurred), invoiced yes/no (if the failure event is invoiced as it is not included within the maintenance agreement with the company, or not).

However, the company also has some in-system data. This in-system data is logging data that the System of Interest (SOI), i.e., APS stores automatically. This logging data includes the status of each subsystem, its position, and the date and time for this status. The in-system data also includes alarm log data. The alarm log data register only the abnormal situation of subsystems, with its position, date, and time.

This abnormal status or operation can be a gate not closed, or a motor may have stopped during the operation. The company has different installations for the APSs and storing mechanisms for each system; some are similar, and others differ. The APS Company cooperates with a third party for their in-system data, as this third party is responsible for this data. Unfortunately, this data can be saved daily or for a few days from only one system. However, the company is investigating if this data can be extracted from some of its systems for a more extended period with the third party. The company has more than 36 installations. The company uses a sheet in the excel file for each system or installation to store their failure events (data). The template for each sheet differs between sheets. This difference includes rearranging the parameters (columns) and describing the same issue using different terminology.

2) Observations at the Oil and Gas Company: : Observations, interviews, and co-creation sessions at the oil and gas company also revealed that they also have unstructured data with some variations of structured data. The unstructured data



In-depth Interview Guide

A. Introduction

1. Consent of Recording

B. Personal Information

1. Name and Experience
2. Work tasks and responsibilities

C. General Questions about Big Data Needs

1. What could be the definition of internal stakeholders? what's your opinion?

2. What could be the internal stakeholders needs while they are using data? According to your experience, is it important to identify internal stakeholder needs?

3. Is the usage of data equally important for all team members? If it is important, then How would be it?

4. What is big data? What kind of data you are using in this project? are you considering this part of big data or just data?

5. Do you think, using data helps to improve products and minimize the production costs, risks? Why do you think this?

6. Would your product or development be better if you had more data?

7. Do you think, using data helps to improve products and minimize the production costs, risks? Why do you think this?

8. Do you have easy access to a set of useful data-analysis tools and methodologies that can help a team to better understand the project task? What types of tools you have used, or you are using?

9. Did you have the similar project as FlexLink project you have now?

10. In your present project, do you have any relevant data from past projects that you think you can use in this project? what kind of relevant data there are? how useful is the relevant data from earlier projects?

11. Do you have access of past data from previous projects? If there is any data, then what kinds of data? How are you (or team) using past data in your current project?

12. If you do not have access to certain data, where could you find it?

13. Can you spend sufficient time to collect and analyse all available data in the early phase? Why you cant spend more time on it?

14. You are doing mobile robot for FlexLink. May be after 10 or 15 years, someone from your group or your company will get the chance to build similar kind of robot. Would be efficient for him if he could have the access of this data and can use this data in his project? How could you ensure this process-do you have any comment?

15. Why is big data important for early-phase development-process? Is there any importance of big data in your project?

16. Is there any procedure for storing previous and current data that you or your team can easily use in your current project? What is the procedure? Why you (don't) have this storing procedure?

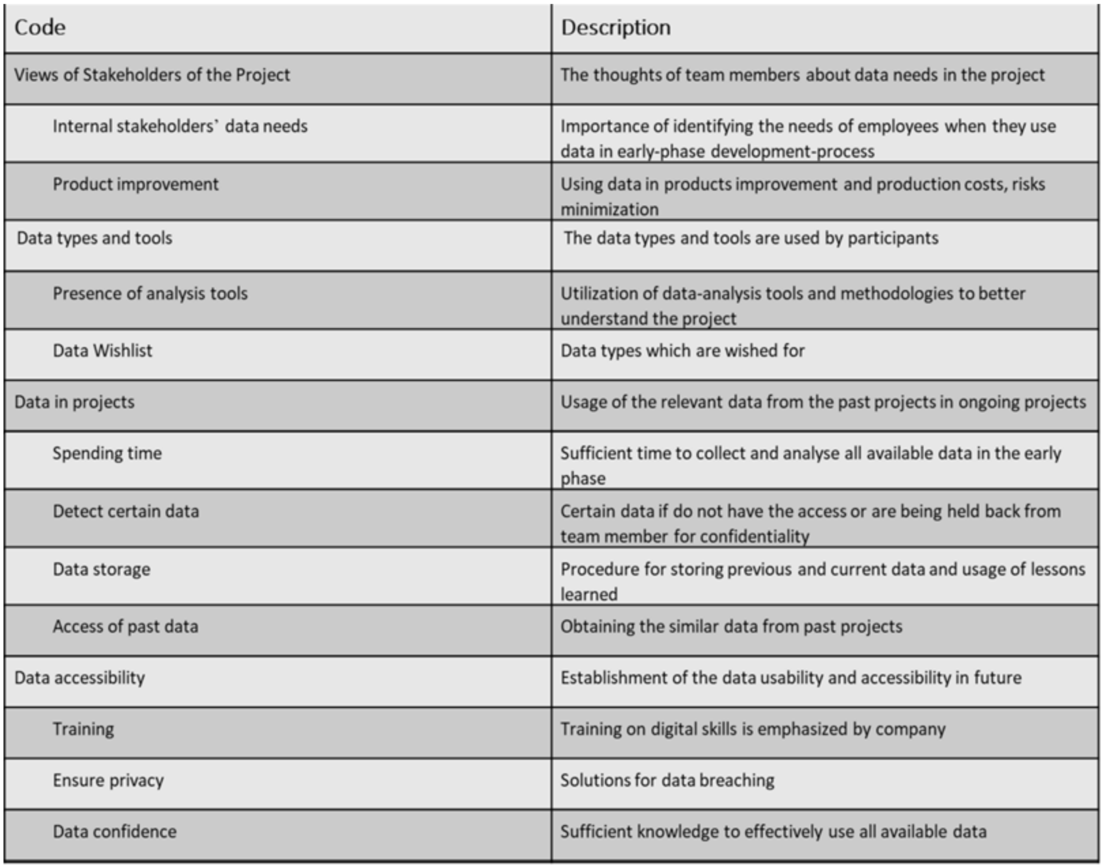
17. Sometimes data and information are being held back from team member because of confidentiality. Have you ever faced this issue in your this project or in your previous project? How will/did you solve this?

18. If data is breached, what (technical) solutions can be taken to ensure privacy?

19. When you work in the early-phases of product-development, are you fully confident that your company (and project team) has sufficient knowledge to effectively use all available data? If no, then which causes behind these problems?

20. How do your company encourage competency-development and training on digital skills such as big data and digitalization? If you do any courses, are these paid and endorsed by the company? Does the company encourage the employees to take these courses?

Fig. 9. Interview Guide.



Code	Description
Views of Stakeholders of the Project	The thoughts of team members about data needs in the project
Internal stakeholders' data needs	Importance of identifying the needs of employees when they use data in early-phase development-process
Product improvement	Using data in products improvement and production costs, risks minimization
Data types and tools	The data types and tools are used by participants
Presence of analysis tools	Utilization of data-analysis tools and methodologies to better understand the project
Data Wishlist	Data types which are wished for
Data in projects	Usage of the relevant data from the past projects in ongoing projects
Spending time	Sufficient time to collect and analyse all available data in the early phase
Detect certain data	Certain data if do not have the access or are being held back from team member for confidentiality
Data storage	Procedure for storing previous and current data and usage of lessons learned
Access of past data	Obtaining the similar data from past projects
Data accessibility	Establishment of the data usability and accessibility in future
Training	Training on digital skills is emphasized by company
Ensure privacy	Solutions for data breaching
Data confidence	Sufficient knowledge to effectively use all available data

Fig. 10. Thematic Categories

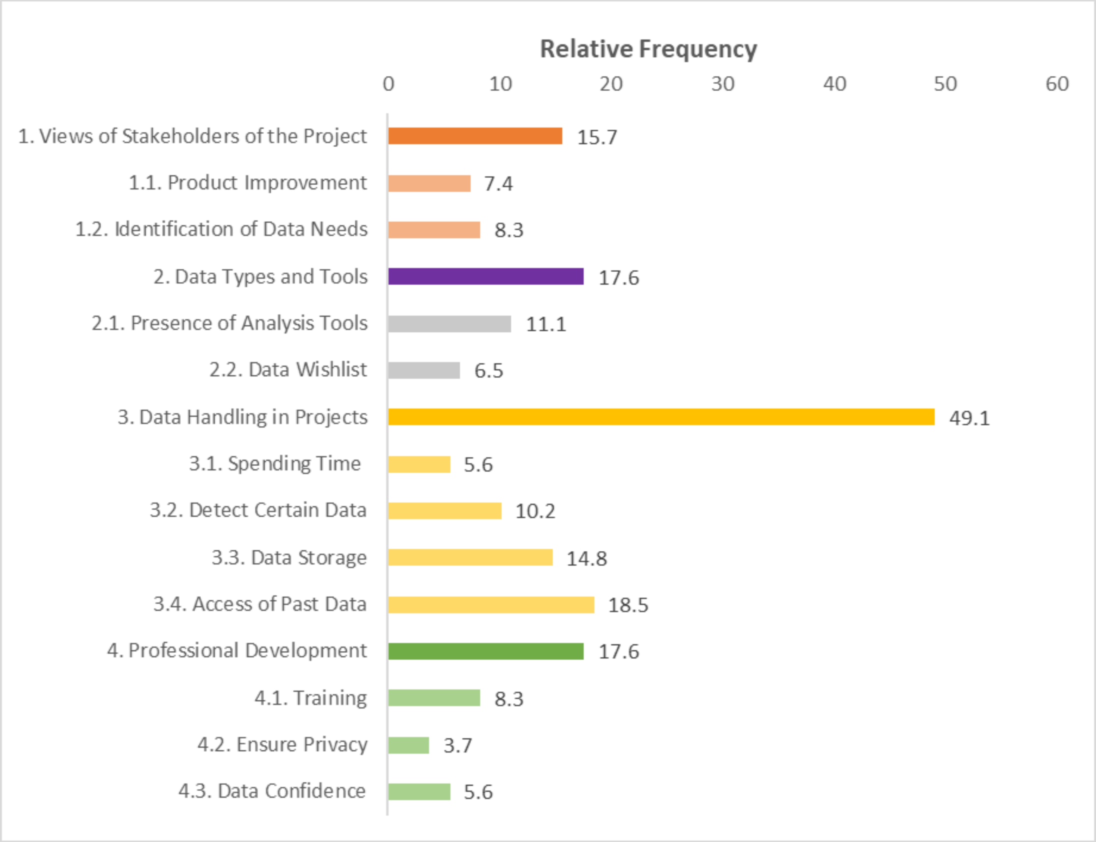


Fig. 11. Visual representations of the categories identified from the interviews. The relative frequencies of the categories and subcategories are also shown in the illustration.

is mainly in event log. The test personnel are logging manually using Excel. The central part of this log is a description of the unexpected events/issues/problems (also called emergent behavior) during the test process, such as the System Integration Test (SIT).

The event log also contains other parameters (columns) such as data about the equipment and its serial number, information about the project, e.g., work package and product responsible department or supplier, and other project-specific information. The company uses an Excel file for each project. However, the template varies for each project. This variation can be rearranging some columns (parameters) and having some rows in the middle of the Excel file. In some Excel files, different terms describe the same issue. There are some errors in the titles of some parameters.

3) Observations at Industrial Conglomerate (Defence Division): We have observed that the defense division struggles with adopting suitable methods for harvesting and utilizing data relevant to Human Systems Integration in unmanned vehicles. A particular area is generating enough appropriate data for supporting designers in exploring and testing various Human Machine Interactions (HMI) solutions. Their decision-making is primarily based on customer requirements, standards, in-house expertise, and subjective opinions from multiple developers. However, they wish to put more emphasis and weight on objective data in their decision-making process. Best practices, such as the Design Thinking process, highlights the need for gaining grounded understanding from relevant users, obtained through prototyping and testing.

The data they need are sourced from the end-users, the components of the SOI, and the system's use. The data can be accessed from the technical system, which is being developed and tested in-house. The defense division also has access to general information on how the system should be used. However, the procedure of how the system should be used is not necessarily the way the system will be used in the field. The organization cannot generate enough data for making objective data based decisions as they have an incomplete picture of the system context, specifically the data from the end-users. As they lack access to all system context information, they can only measure and analyze their data, not necessarily the data they should analyze.

V. Concluding Remarks

The case study gave us a deeper understanding of the state of affairs at some of the industry partners. It provided a glimpse at the disparity of different organizations regarding their utilization of big data in their operations and decision-making process, focusing on the early phase product development process. Overall, the case study concluded that employ-

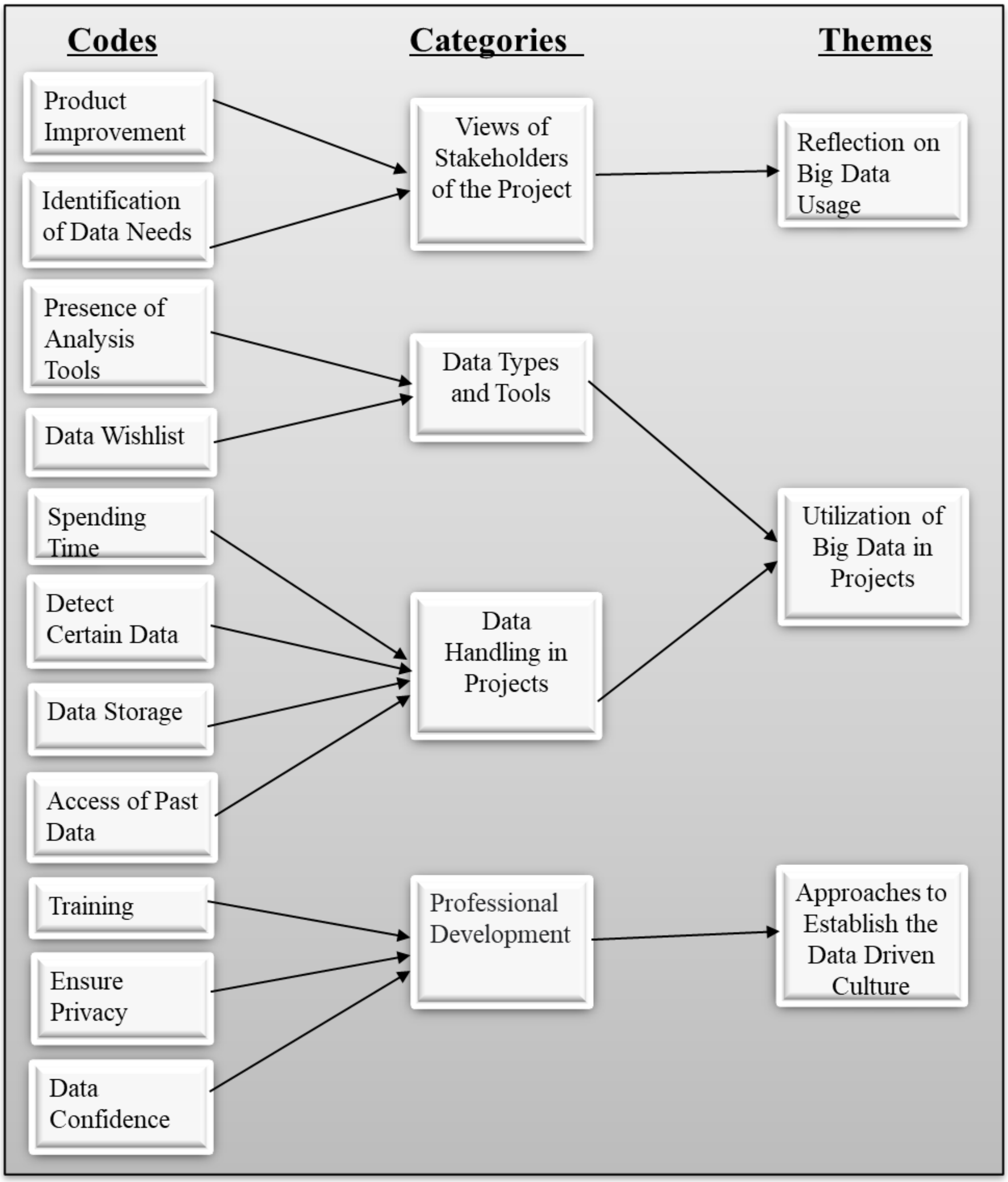

Codes
Categories
Themes
Product Improvement
Identification of Data Needs
Presence of Analysis Tools
Data Wishlist
Spending Time
Detect Certain Data
Data Storage
Access of Past Data
Training
Ensure Privacy
Data Confidence
Views of Stakeholders of the Project
Data Types and Tools
Data Handling in Projects
Professional Development
Reflection on Big Data Usage
Utilization of Big Data in Projects
Approaches to Establish the Data Driven Culture

Fig. 12. Themes used for the thematic analysis of the conversation with employees.

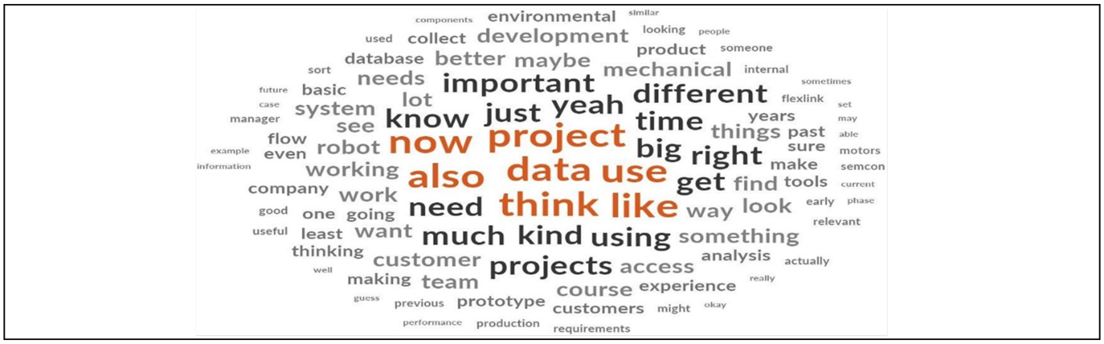


Fig. 13. Word cloud showing the most common words appears in the “Reflection on Big Data Usage” theme.

ees at the H-SEIF2 industry partners understand the need to use big data in their projects to enhance their operations.

The employees at the technology consultancy reported that a considerable amount of data generated after every project is often stored only locally by the personnel instead of stored in an accessible database of the company. The company lacks sophisticated big data analysis tools to understand a complex project thoroughly. Currently, the company uses Python, Excel, and similar common tools for simple data analysis.

Similarly, in the observations at both Automated Parking System and oil and gas provider, we observed that both companies use Excel files that either maintenance or test personnel log manually. However, the template for the Excel file differs slightly for each system or project. This difference makes it cumbersome to automate the pre-processing in different degrees depending on the data, especially when gathering historical data for an extended period, e.g., five or ten years. Thus, manual pre-precessing is needed. In other words, we must pre-process the data manually by generating a template called frame around the data. This manual pre-processing consumes almost 80% of the analysis period.

However, manual pre-precessing is time-consuming and may result in some errors when doing it manually. Therefore, we recommend that the company unify the template and use a not-editing version. Also, only certain manager-level employees should change the template, not everyone in the test or maintenance department. In [49], we suggested a template that integrates the needed data and information on one platform using one tool.

For the defense division, the main limitation is a lack of end-user data. There are primarily two reasons for such lacking. Firstly, they have no access to the end-users usage in field operations due to security reasons. They are building simulators to combat limited access to end-users and end-user data. Secondly, they lack integrated and developed test procedures for simulator testing to generate and collect human factors data. The human factor data include physiological, mental, and operational data. These data can be collected through, for example, eye tracking, heart and galvanic skin response measurement data to measure the stress of the end-users, interviews and test questions, and performance data. These data aim is to understand the HMI influence on situational awareness during operations.

The presented case study provided the current state of big data usage scenarios using different research methods at Norwegian high-tech companies and identified issues hindering big data's full potential. The observations from the case study can be used for future research on similar business organizations in addressing the internal stakeholders' needs regarding big data usage.

REFERENCES

[1] F. A. Salim, H. B. Ali, T. Langen, A. Wettre, G. Muller, and K. Falk, "State of Affair in Terms of Big Data Utilization in Complex System Engineering Organizations," in *MODERN SYSTEMS 2022: International Conference of Modern Systems Engineering Solutions*, no. c, Nice, France, 2022, pp. 45–49.

[2] R. K. Perrons and J. W. Jensen, "Data as an asset: What the oil and gas sector can learn from other industries about "Big Data"," *Energy Policy*, vol. 81, pp. 117–121, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.enpol.2015.02.020

[3] R. Technology, "AI / ML Annual Research Report 2022," 2022, (last accessed 05-Dec-2022). [Online]. Available: https://mikenashtech.com/2022/02/ai-ml-annual-research-report-2022-analysis/

[4] K. Falk, A. K. Kamara, E. P. Brathen, K. Helle, P. T. Moe, and S. Kokkula, "Digitizing the Maintenance Documentation; A System of Systems in Oil and Gas Industry," *SOSE 2020 - IEEE 15th International Conference of System of Systems Engineering, Proceedings*, pp. 493–499, 2020.

[5] H. B. Ali and F. A. Salim, "Transferring Tacit Knowledge into Explicit: A Case Study in a Fully (Semi) Automated Parking Garage," in *PROCEEDINGS OF The Society for Design and Process Science*, 2021.

[6] F. A. Salim, H. B. Ali, and K. Falk, "Towards a Data Processing Framework for Enhanced User Centric System Engineering," in *PROCEEDINGS OF The Society for Design and Process Science, Workshop on Smart Pervasive Computing*, 2021.

[7] E. Raguseo, "Big data technologies: An empirical investigation on their adoption, benefits and risks for companies," *International Journal of Information Management*, vol. 38, no. 1, pp. 187–195, 2018.

[8] F. A. Salim, F. Haider, S. Luz, and O. Conlan, "Automatic transformation of a video using multimodal information for an engaging exploration experience," *Applied Sciences (Switzerland)*, vol. 10, no. 9, 2020.

[9] Qlik and Accenture, "The Human Impact of Data Literacy," Tech. Rep., 2020, (last accessed 05-Dec-2022). [Online]. Available: https://thedataliteracyproject.org/humanimpact

[10] B. A. Delicado, A. Salado, and R. Mompó, "Conceptualization of a T-Shaped engineering competency model in collaborative organizational settings: Problem and status in the Spanish aircraft industry," *Systems Engineering*, vol. 21, no. 6, pp. 534–554, 2018.

[11] S. Engen, K. Falk, and G. Muller, "The need for systems awareness to support early-phase decision-making—a study from the norwegian energy industry," *Systems*, vol. 9, no. 3, 2021.

[12] Juzgado P.D. Borches, "A3 Architecture overviews. A tool for effective communication in product evolution," Ph.D. dissertation, University of Twente, 2010.

[13] T. Tomiyama, V. D'Amelio, J. Urbanic, and W. Eimaraghy, "Complexity of multi-disciplinary design," *CIRP Annals - Manufacturing Technology*, vol. 56, no. 1, pp. 185–188, 2007.

[14] U. NISE, "H-SEIF 2 Project," (last accessed 05-Dec-2022). [Online]. Available: https://www.usn.no/hseif

[15] C. Potts, "Software-engineering research revisited," *IEEE Software*, vol. 10, pp. 19–28, 1993.

[16] M. Guntveit, M. Kjrstad, and B. Sevaldson, "Early Validation of Stakeholder Needs by Applying Co-creation Sessions," *INCOSE International Symposium*, vol. 30, no. 1, pp. 1398–1415, 2020.

[17] G. Gravier, M. Ragot, A. Laurent, R. Bois, G. Jadi, E. Jamet, and L. Monceaux, "Shaping-Up Multimedia Analytics: Needs and Expectations of Media Professionals," in *The 22nd International Conference on Multimedia Modelling*, vol. 9516, Miami, 2016, pp. 303–314.

[18] M. Haesen, J. Meskens, K. Luyten, K. Coninx, J. H. Becker, T. Tuytelaars, G. J. Poulisse, P. T. Pham, and M. F. Moens, "Finding a needle in a haystack: An interactive video archive explorer for professional video searchers," *Multimedia Tools and Applications*, vol. 63, no. 2, pp. 331–356, 2013.

[19] F. A. Salim, F. Haider, O. Conlan, and S. Luz, "An approach for exploring a video via multimodal feature extraction and user interactions," *Journal on Multimodal User Interfaces*, no. October 2017, 2018.

[20] O. Kwon and J. Sim, "Effects of data set features on the performances of classification algorithms," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1847–1857, 2013.

[21] P. Russom, "Big data analytics," *TDWI best practices report*, vol. 19, no. 4, pp. 1–34, 2011.

[22] D. Laney, "3D data management: Controlling data volume, velocity and variety," *META group research note*, vol. 6, no. 70, 2001.

[23] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," *IDC iView: IDC Analyze the future*, vol. 2007, pp. 1–16, 2012.

[24] J. Dijcks, "Oracle: Big data for the enterprise," Oracle, Tech. Rep., 2012.

[25] S. Gogia, M. Barnes, B. Evelson, B. Hopkins, N. Yuhanna, D. Anderson, and H. Kisker, "The Big Deal About Big Data For Customer Engagement," Forrester, Tech. Rep., 2012, (last accessed

05-Dec-2022). [Online]. Available: https://www.forrester.com/report/The-Big-Deal-About-Big-Data-For-Customer-Engagement/RES72241

[26] M. White, "Digital workplaces: Vision and reality," *Business information review*, vol. 29, no. 4, pp. 205–214, 2012.

[27] H. B. Ali, F. H. Helgesen, and K. Falk, "Unlocking the power of big data within the early design phase of the new product development process," *INCOSE International Symposium*, vol. 31, no. 1, pp. 434–452, 2021.

[28] B. Schätz, M. Törngren, R. Passerone, S. Bensalem, A. Sangiovanni-Vincentelli, J. McDermid, H. Pfeifer, and M. Cengarle, "CyPhERS-cyber-physical European roadmap and strategy," fortiss GmbH, Munich, Tech. Rep., 2014.

[29] A. Alharthi, V. Krotov, and M. Bowman, "Addressing barriers to big data." *Business Horizons*, vol. 60, no. 3, pp. 285–292, 2017.

[30] M. Douglas, "Big data raises big questions," *Government Technology*, vol. 26, no. 4, pp. 12–16, 2013.

[31] S. Hoffman and A. Podgurski, "Big bad data: law, public health, and biomedical databases," *Journal of Law, Medicine Ethics*, vol. 41, no. 1, pp. 56–60, 2013.

[32] B. Laugwitz, T. Held, and M. Schrepp, "Construction and Evaluation of a User Experience Questionnaire," *HCI and Usability for Education and Work*, pp. 63–76, 2008.

[33] G. Nicolini and L. D. Valle, "Errors in Customer Satisfaction Surveys and Methods to Correct Self-Selection Bias," *Quality Technology Quantitative Management*, vol. 8, no. 2, pp. 167–181, 2011.

[34] B. Popovic, R. Maletic, and T. Paunovic, "Employee Satisfaction Survey in Function of Business Improvement," *Management - Journal for theory and practice of management*, vol. 20, no. 76, pp. 31–40, 2015.

[35] L. She, L. Ma, A. Jan, H. Sharif Nia, and P. Rahmatpour, "Online Learning Satisfaction During COVID-19 Pandemic Among Chinese University Students: The Serial Mediation Model," *Frontiers in Psychology*, vol. 12, no. October, 2021.

[36] B. Dudley, "Digital Transformation Initiative: Oil and Gas Industry In collaboration with Accenture," *World Economic Forum*, no. January, p. 32, 2017, (last accessed 05-Dec-2022). [Online]. Available: www.weforum.org

[37] D. H. Shin, "Demystifying big data: Anatomy of big data developmental process," *Telecommunications Policy*, vol. 40, no. 9, pp. 837–854, 2016.

[38] G. Muller, "Systems engineering research methods," *Procedia Computer Science*, vol. 16, pp. 1092–1101, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2013.01.115

[39] ——, "Tutorial Architectural Reasoning Using Conceptual Modeling," University of South-Eastern Norway, Kongsberg, Tech. Rep., 2015. [Online]. Available: https://www.gaudisite.nl/TutorialARconceptualModelingSlides.pdf

[40] I. Seidman, *Interviewing as qualitative research: A guide for researchers in education and the social sciences:*, 3rd ed. Teachers college press, 2006.

[41] R. Berry, "Collecting data by in-depth interviewing," in *British Educational Research Association Annual Conference*, Brighton, 1999.

[42] M. Saunders, P. Lewis, and A. Thornhill, *Research methods for business students*. Pearson education, 2009.

[43] N. M. Sjøkvist and M. Kjørstad, "Eliciting Human Values by Applying Design Thinking Techniques in Systems Engineering," *INCOSE International Symposium*, vol. 29, no. 1, pp. 478–499, 2019.

[44] V. Braun and V. Clarke, "Using thematic analysis in psychology." *Qualitative research in psychology*, vol. 3, no. 2, pp. 77–101, 2006.

[45] J. Anuradha, "What is the Difference Between Thematic and Content Analysis," 2022, (last accessed 05-Dec-2022). [Online]. Available: https://pediaa.com/what-is-the-difference-between-thematic-and-content-analysis/

[46] A. Bogetz, E. Abramson, H. Haftel, and M. Klein, "Codes, concepts and categories, oh my! Building your skills in qualitative data analysis," Association of Pediatric Program Directors, Anaheim, Tech. Rep., 2017. [Online]. Available: https://appd.s3.amazonaws.com/docs/meetings/2017SpringPresentations/WS10Slides.pdf

[47] G. Gibbs, "Thematic coding and categorizing," in *Analyzing qualitative data*. SAGE Publications, Ltd, 2007, ch. 4, pp. 38–56.

[48] J. Corbin and A. Strauss, *Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory*, 4th ed. SAGE Publications, Ltd, 2014.

[49] F. A. Salim, H. B. Ali, G. Muller, and K. Falk, "User-centered Data Driven Approach to Enhance Information Exploration, Communication and Traceability in a Complex Systems Engineering Environment," in *MODERN SYSTEMS 2022: International Conference of Modern Systems Engineering Solutions*, no. c, 2022, pp. 37–42.

Optimizing Remediation of Spatially Dispersed Contaminated Parcels under an Annual Budget Constraint

Floris Abrams[1,2], Lieve Sweeck[1], Johan Camps[1]
[1]Belgian Nuclear Research Centre (SCK CEN),
Mol, Belgium
e-mail: {Floris.Abrams, Lieve.Sweeck,
Johan.Camps}@sckcen.be

Dirk Cattrysse[2], Jos Van Orshoven[2]
[2]Katholieke Universiteit Leuven (KU Leuven)
Leuven, Belgium
e-mail: {Dirk.Cattrysse, Jos.Vanorshoven}@kuleuven.be

***Abstract*—In environmental disaster management, due to the large impacted area or limited availability of labor and financial resources, setting priorities of where, how and when to act are indispensable. When prioritized interventions on spatially dispersed entities are costly and technically challenging to perform, clustering of individual entities in larger homogeneous actionable units can improve feasibility and reduce cost of the remediation. In this article, a spatio-temporal clustering approach under a budget constraint is presented to determine homogenous clusters of polygons and interventions to reduce cost while still attaining an overall optimal distribution of interventions. We demonstrate the effectiveness of this clustering algorithm with a hypothetical case study of contaminated agricultural land in Belgium. Finally, we demonstrate the capabilities of the proposed cluster algorithm to provide decision makers with a multi-period action plan, reducing the cost of intervention while still prioritizing resources for the most important sites.**

Keywords-Spatio-temporal clustering; Budget constraint; Disaster management; Multi-Attribute Decision Making; MADM.

I. INTRODUCTION

This paper extends a previous paper that was originally presented at the Fourteenth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing) [1].

When dealing with large natural or man-made disasters, decision makers are confronted with setting priorities of where, how and when to act because of the limited availability of labor and financial resources. This priority setting is particularly applicable when the impact of remedial actions is costly and has long-lasting influences. For spatially distributed sites with variable characteristics, priority setting among the sites and the determination of the most adequate remedial action per site are of major importance. The United States Environmental Protection Agency (US EPA) identified the following benefits of these optimization efforts: more cost-effective expenditure, lower energy use, reduced carbon footprint, improved remedy protectiveness, improved project and site decision making, and acceleration of project and site completion [2].

Addressing the questions of where and how to act consecutively results in a nested ranking of sites and interventions per site. From those rankings, a spatio-temporal action plan can be determined.

To assist decision makers in setting such priorities, spatial Decision Support Systems (sDSS) become of importance [3]. The effectiveness of related decisions is typically conditioned by multiple and often contradicting criteria of economic, social, technical, environmental, and human health-related nature [4]. These characteristics of the decision problem make it suitable for the application of a spatially discrete Multi-Attribute Decision Making (GIS-MADM) approach [5]. 'GIS' points to the spatial aspect of the decision problem, while MADM encompasses a subset of Multi-Criteria Decision Analysis (MCDA) methods. MADM supports the decision-maker by describing and evaluating the performance of a finite number of decision alternatives with respect to multiple criteria expressed as attributes of the alternatives, representing several points of view. The MADM results in a ranking of the alternatives based on the selected criteria and their relative importance [3]. The MADM framework is often applied because it supports a structured and inclusive decision process, addressing a plurality of preferences and socio-technical dimensions that cannot always be brought to a common monetary scale [6].

This paper presents a GIS-MADM approach that provides actionable support to decision makers by proposing a coherent action plan in space and time for decontamination of the agricultural domain in a region affected by the deposition of radionuclides. It uses a spatio-temporal approach to deal with the clustering of spatially scattered polygon-based parcels, whereby a budget constraint limits the extent and/or type of interventions that can be performed in each time step, i.e., in one year. The paper elaborates on the classic region-growing principles, adapted to polygon-based data structures, and explicitly takes into account the attributes of the individual polygons to find the optimal compromise attribute for the whole cluster. Because a spatial and temporal clustering of sites and actions is likely to create "economies of scale" [7], the cost of remediation interventions will be lowered, resulting in an overall cheaper and faster remediation process.

The rest of this paper is organized as follows: Section II presents related work where MADM is used to support prioritization of resources in environmental remediation.

Section III provides an in depth explanation of the spatio-temporal cluster approach. In Section IV, the approach is illustrated with a case study for an agricultural region in Belgium, contaminated after a hypothetical accidental release of Caesium-137 from a nuclear power plant. Section V discusses the applicability of the algorithm to help improve decision making, while Section VI draws the most pertinent conclusions.

II. Related Work

The use of MADM approaches for supporting remediation on a regional scale by prioritization contaminated sites for decontamination (‘Where to act?’) was reported by several authors [8]–[10]. In addition, the support on a local scale by prioritization of the remedial technologies for a given site (‘How to act?’) was also addressed in several publications [11]–[13]. However, no reports were found, where MADM was used for simultaneously prioritizing of where and how to act decisions into a coherent spatio-temporal action plan. When both prioritizations are done separately, the procedure typically yields a geographically distributed set of priority sites as well as neighbouring sites with different interventions. Different propositions were made to improve MADM on a regional scale, to reduce the scattered priorities. For example, by incorporating a compactness measure to ensure sites were big enough to ensure a feasible intervention [14].

MADM approaches have been used with raster as well as polygon-based datasets. For this application, it was chosen to use polygon-based data because they provide a natural representation for many types of geospatial entities, such as agricultural parcels, buildings, or polluted sites. In addition, these entities form the smallest units used in real-world decision making. Therefore, it is interesting to provide actionable support to decision makers based on polygon-based representations. By addressing the problem with polygon-based data, the adaptations using compactness measures and clustering of entities are more complicated compared to raster-based datasets. Because the topology of spatially dispersed polygons is less straightforward when dealing with unlinked features [15].

Further, due to the limited availability of resources, a budget constraint limits the extent of interventions possible for each period; thus, a multiple-period action plan is required. A multi-period decision problem requires a Dynamic Multi-Attribute Decision Making (dMADM) methodology [17]–[19]. However, the majority of documented MADM applications only address a decision problem for a specific time period [16]. In contrast, dMADM determines the criteria scores and relative relevance for each time period to accurately reflect the decision variables at that time.

Some spatial DSS tools were developed for supporting decisions with similar spatio-temporal aspects. These authors included a temporal dimension in their approach to determine how a set of land use types should be distributed over space and time in order to optimize the multi-dimensional land performance of a region over a period of 30 years [20]. However, they found that their approach, which was based on integer programming (IP), resulted in land use plans that were too spatially and temporally fragmented for real-world application and recommended that a clustering strategy could be a suitable next step.

III. Proposed Method

The spatio-temporal clustering approach combines a site priority score (PPS) and an action priority score (APS), as discussed in Section A. The iterative and dynamic cluster growing algorithm is discussed in Sections B and C.

A. Distance based priority scores

Different implementations of MADM exist, each with their own strengths and weaknesses. We opted for a distance-based MADM, called Compromise Programming (CP), to rank the considered set of feasible alternatives [21]–[24]. CP uses the distance in the feature space to the so-called ideal point of each alternative to rank them. The feature space is constructed from independent, operational, non-redundant, and continuous attributes [25]. The criteria used vary significantly between different case studies, depending on the problem, the site characteristics and the available data. For each criterion, a weight reflecting the importance of the criterion is set by the stakeholders, preferably through a collaborative process [26]. This weight takes into account the relative importance of the criterion, where its value can be understood as a trade-off value between criteria. For this set of criteria and corresponding weights, the CP methodology determines the optimal point, a vector of performance attribute values corresponding to an alternative with the best observed performance on each criterion separately. This ideal point is mostly hypothetical, because multi-criteria decision problems involve conflicting criteria. The ideal point does however allow to determine a ranking of the alternatives based on each alternative’s distance to the ideal point, whereby the alternative that comes ‘closest’ to the ideal point is the most preferred. The definition of ‘closeness’ requires the formulation of a distance metric (1), where a larger distance equals a less optimal alternative [3]. Distances based on (1) fall within the range [0-1], with a distance of 0 being the best alternative that requires no compromise because it outperforms all other alternatives on all criteria. In contrast, a distance of 1 reflects an alternative that scores the lowest on all criteria.

$$L = \left[\sum_{i}^{n} \left[\frac{f_i^+ - f_i(x)}{f_i^+ - f_i^-} \right]^p \right]^{1/p} \qquad (1)$$

– n is the number of criteria under consideration;
– w_i is the relative importance (weight) assigned to performance attribute i;

- p is a parameter that determines the type of distance function, where 2 represents the Euclidian distance;
- f_i^+ is the optimal value for performance criterion i;
- $f_i(x)$ is the value of the i^th^ performance criterion expressed as a function of the decision variables x;
- f_i^- is the anti-ideal corresponding to the ith attribute that is the "worst" value for this attribute.

To determine the optimal remediation plan for a territory of interest two important questions need to be answered. The first question is "Where are the sites situated for which intervention is most urgent?". The CP methodology returns a distance score for each polygon, representing the priority/urgency of a polygon to be intervened on. From these scores a ranking of the polygons from high priority (small distance) to low priority (large distance) can be made. For the case study in this paper, this score is referred to as Parcel Priority Score (PPS). The second question is "What is the most optimal action for each site?". Therefore, for each polygon, the feasible intervention actions need to be ranked. In our proposed approach, the ranking of the alternative interventions is similarly based on a distance score, computed by CP. In the following case study, for each alternative intervention on a specific site, the Action Priority Score (APS) is calculated. The further clustering of parcels is based on the combination of PPS and APS.

B. *Temporal dynamics in MADM*

When actions are postponed in time, the initial decision variables (criteria scores for the alternatives and criteria weights) may alter, and the decision problem needs to be redefined, resulting in a multi-period MADM. The number and extent of polygons that can be acted on in each time period depends on the budget available in each period, which is set to one year in our case study. While performing actions on the most urgent polygons first, each of the actions comes at a cost. For each intervention, the cost can be calculated based on the cost per unit of area and the size of the polygon. Interventions can be done in one period until the total cost of remediation exceeds the period's budget. When the budget is reached, the remaining polygons become candidates for the next period, where changes in the criteria scores and weights may occur and should be taken into account.

C. *Spatio-temporal clustering algorithm*

The algorithm operates in a similar fashion as a region-growing algorithm, where it consecutively checks whether one of the neighbouring polygons can be added to the cluster, taking the similarity between the priority scores of the seed polygon and the neighbouring candidate into account. The clustering algorithm is iterative and consists of two phases: The cluster initialization phase is followed by the cluster growing phase, which ends as soon as one of the stopping criteria is met. The procedure is illustrated in Figure 1 and the pseudo code is given in Figure 2.

1) Cluster initialisation

To optimally allocate resources, the most urgent sites should be treated first. Therefore, the seed parcel is the one with the lowest PPS (smallest distance to the ideal point).

2) Cluster growing procedure

After the seed parcel has been determined, the cluster-growing procedure attempts to find neighbouring parcels that can be added to the seed parcel or the growing cluster, where parcels in a cluster have the same intervention action to be performed in the same period.

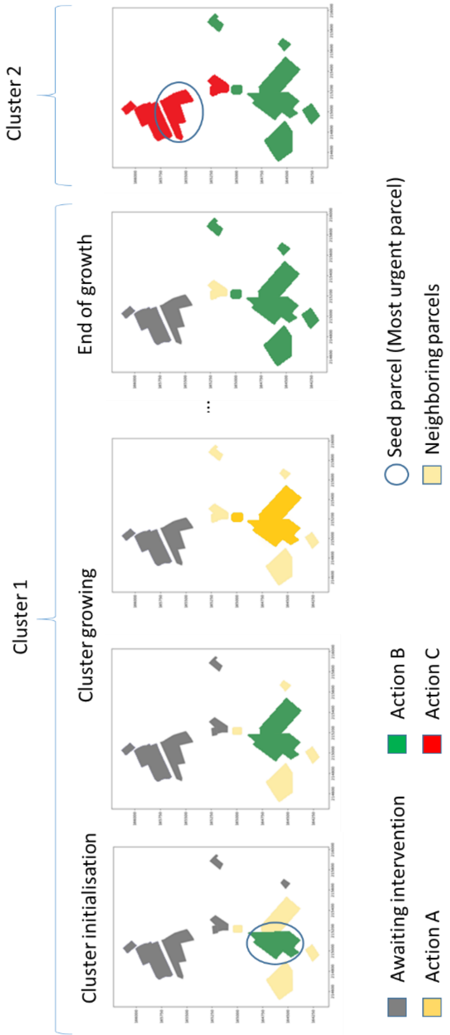
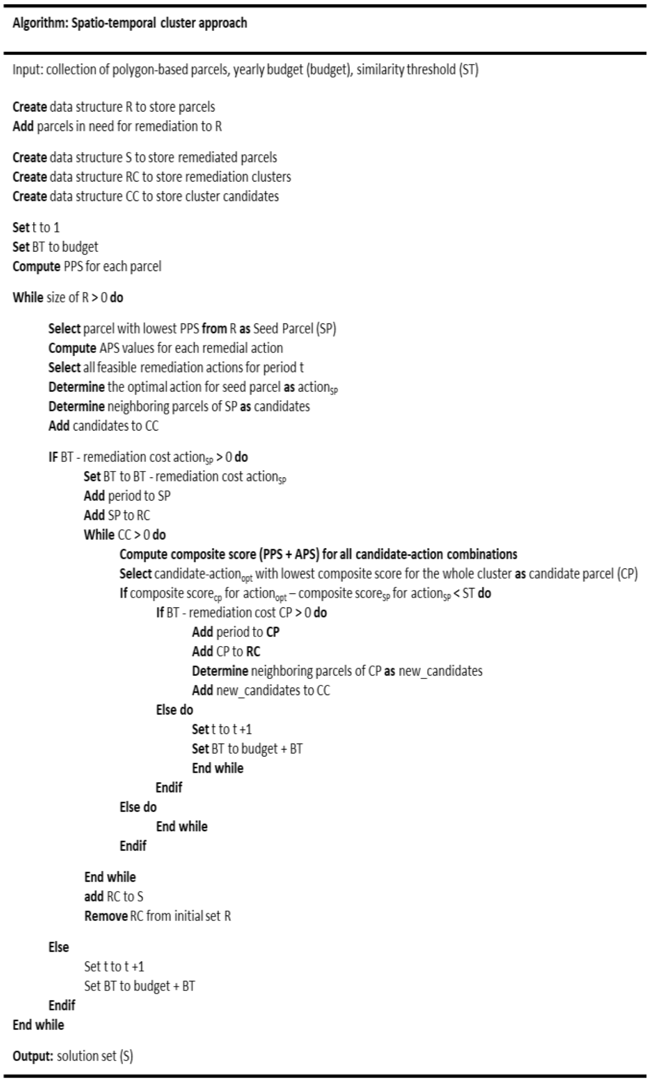


FIGURE 1. THE CLUSTER GROWING PROCEDURE APPLIED ON 12 PARCELS, CONSIDERING 3 POSSIBLE ACTIONS, RESULTING IN 2 CLUSTERS EACH WITH 1 ACTION.



```
Algorithm: Spatio-temporal cluster approach

Input: collection of polygon-based parcels, yearly budget (budget), similarity threshold (ST)

Create data structure R to store parcels
Add parcels in need for remediation to R

Create data structure S to store remediated parcels
Create data structure RC to store remediation clusters
Create data structure CC to store cluster candidates

Set t to 1
Set BT to budget
Compute PPS for each parcel

While size of R > 0 do

    Select parcel with lowest PPS from R as Seed Parcel (SP)
    Compute APS values for each remedial action
    Select all feasible remediation actions for period t
    Determine the optimal action for seed parcel as action$_{sp}$
    Determine neighboring parcels of SP as candidates
    Add candidates to CC

    IF BT - remediation cost action$_{sp}$ > 0 do
        Set BT to BT - remediation cost action$_{sp}$
        Add period to SP
        Add SP to RC
        While CC > 0 do
            Compute composite score (PPS + APS) for all candidate-action combinations
            Select candidate-action$_{opt}$ with lowest composite score for the whole cluster as candidate parcel (CP)
            If composite score$_{cp}$ for action$_{opt}$ – composite score$_{sp}$ for action$_{sp}$ < ST do
                If BT - remediation cost CP > 0 do
                    Add period to CP
                    Add CP to RC
                    Determine neighboring parcels of CP as new_candidates
                    Add new_candidates to CC
                Else do
                    Set t to t +1
                    Set BT to budget + BT
                    End while
                Endif
            Else do
                End while
            Endif

        End while
        add RC to S
        Remove RC from initial set R

    Else
        Set t to t +1
        Set BT to budget + BT
    Endif
End while

Output: solution set (S)
```

FIGURE 2. PSEUDO CODE OF THE SPATIO-TEMPORAL CLUSTER APPROACH, DETERMINING THE REMEDIAL TECHNIQUE AND TIMING OF THE CLUSTERS.

Adding more parcels to the cluster enlarges the cluster, therefore creating larger actionable units, which are preferred from the perspective of reducing the complexity and operational cost of the intervention. But since the parcels added to the cluster potentially have a different optimal action, it is important to find a compromise remediation action that minimizes the deviation in performance with the parcels considered individually. The cluster growing can be subdivided into three consecutive steps that are repeated until the constraints for the end of cluster growth are met.

a) *Determination of the parcel neighbours*

Compared to a raster dataset, where pixels are spatially arranged in a systematic way and neighbours are easily defined, in a data set of spatially distributed polygons, determining the neighbours is more challenging. To define neighbouring polygons, which are not necessarily sharing a border but are rather separated by irrelevant space, a technique called morphologic tessellation (MT) is used. At the core of MT is the Voronoi tessellation (VT), a method of geometric partitioning of the 2D space, where a planar set of "seed points" generates a series of polygons known as Voronoi polygons (VP). Each VP encloses the portion of the plane that is closer to its seed than to any other polygon [27]. From the partitioned space, the neighbours of a VP can be determined by examining the VPs sharing borders. An example of the portioning by VPs is given in Figure 3.

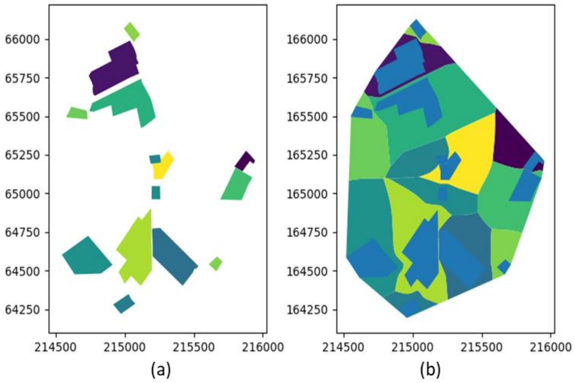


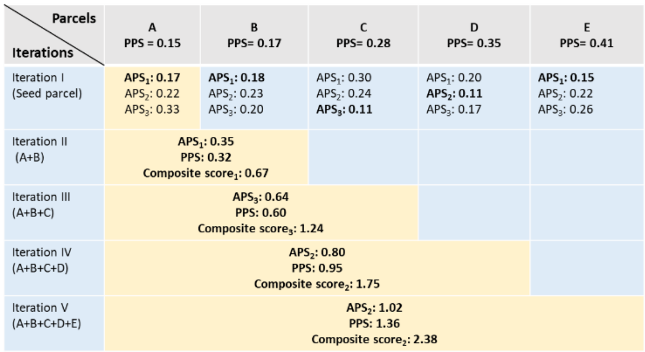
FIGURE 3. INITIAL SET OF DISTRIBUTED PARCELS (A) AND VP COMPUTED BY THE EMT, RESULTING IN A PARTITIONED COVERAGE (B).

To deal with the distributed nature of the polygons, use is made of an enclosed tessellation based on the enhanced morphological tessellation algorithm (EMT). EMT allows for setting limits to the expansion of the MT, limiting the allowed distance between polygons that can be considered to be neighbours. Furthermore, it allows for the establishment of break lines (e.g., rivers or administrative boundaries) beyond which the VPs are not permitted to trespass. The VP constructed by the EMT algorithm captures the spatial configuration of all parcels, from which the neighbouring parcels of each parcel can be determined. The EMT algorithm is accessible from an open-source Python package (http://docs.momepy.org). Fleischmann (2019, 2020) provides more information regarding the EMT methodology.

b) *Determining the optimal neighbour*

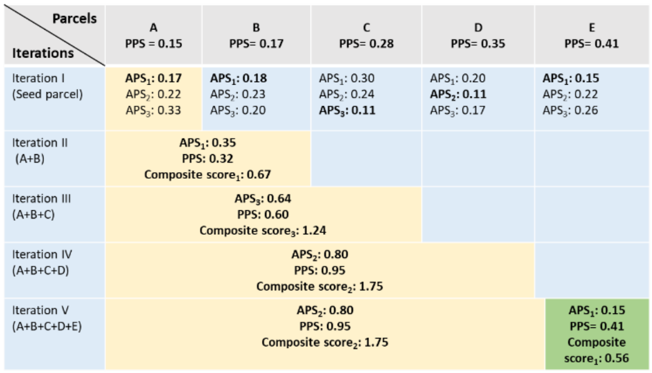
To determine the neighbouring polygon that is best suited for growing the cluster, the sum of the PPS and APS scores of each neighbour is considered. Whereby the neighbour leading to the lowest increase in the composite score of the cluster is added. From this, it follows that adding a parcel to the cluster can change the remediation action to be applied to all the parcels in the cluster. Moreover, when the best candidate is found, it is verified whether the candidate neighbour is similar enough to the seed pixel to be added. If the similarity threshold is not exceeded, the parcel is added to the cluster, and this procedure is repeated; otherwise, the end of the cluster growing phase is reached. To highlight the process of finding a compromise between all parcels on the cluster level, five iterations of the growing procedure are shown in Table 1 and Table 2. The similarity threshold applied is 0.31 for Table 1 and 0.15 for Table 2. The cells with the same color show the current parcels in the cluster, and the remedial action of the cluster is shown with a subscript on the APS. The APS values in bold show the optimal action per parcel. Table 1 illustrates that while a cluster grows iteratively, the optimal remediation action for all parcels combined within the cluster changes. In iteration III, the optimal remediation on the cluster level is the worst-performing action for the seed parcel (A) and the second-best action for parcel B. Nevertheless, from the perspective of the cluster, action 3 is the best compromise solution. In addition, Table 2 shows the impact of the similarity threshold: In iteration V, parcel E is not added to the growing cluster due to a difference larger than the similarity threshold between it and the seed parcel (parcel A). Parcel E will then be selected as the next seed parcel. The different cluster configuration (1 vs. 2) in iteration V for both tables highlights that a lower similarity threshold will result in an overall lower (better) composite score for the solution.

TABLE 1. THE GROWING PROCEDURE OF A CLUSTER FOR 5 ITERATIONS FOR A SIMILARITY THRESHOLD OF 0.3, RESULTING IN ONE CLUSTER



Parcels / Iterations	A PPS = 0.15	B PPS= 0.17	C PPS= 0.28	D PPS= 0.35	E PPS= 0.41
Iteration I (Seed parcel)	**APS$_1$: 0.17** APS$_2$: 0.22 APS$_3$: 0.33	**APS$_1$: 0.18** APS$_2$: 0.23 APS$_3$: 0.20	APS$_1$: 0.30 APS$_2$: 0.24 **APS$_3$: 0.11**	APS$_1$: 0.20 **APS$_2$: 0.11** APS$_3$: 0.17	**APS$_1$: 0.15** APS$_2$: 0.22 APS$_3$: 0.26
Iteration II (A+B)	**APS$_1$: 0.35** **PPS: 0.32** **Composite score$_1$: 0.67**				
Iteration III (A+B+C)	**APS$_3$: 0.64** **PPS: 0.60** **Composite score$_3$: 1.24**				
Iteration IV (A+B+C+D)	**APS$_2$: 0.80** **PPS: 0.95** **Composite score$_2$: 1.75**				
Iteration V (A+B+C+D+E)	**APS$_2$: 1.02** **PPS: 1.36** **Composite score$_2$: 2.38**				

TABLE 2. THE GROWING PROCEDURE OF A CLUSTER FOR 5 ITERATIONS FOR A SIMILARITY THRESHOLD OF 0.15, RESULTING IN TWO CLUSTERS.



Parcels / Iterations	A PPS = 0.15	B PPS= 0.17	C PPS= 0.28	D PPS= 0.35	E PPS= 0.41
Iteration I (Seed parcel)	**APS$_1$: 0.17** APS$_2$: 0.22 APS$_3$: 0.33	**APS$_1$: 0.18** APS$_2$: 0.23 APS$_3$: 0.20	APS$_1$: 0.30 APS$_2$: 0.24 **APS$_3$: 0.11**	APS$_1$: 0.20 **APS$_2$: 0.11** APS$_3$: 0.17	**APS$_1$: 0.15** APS$_2$: 0.22 APS$_3$: 0.26
Iteration II (A+B)	**APS$_1$: 0.35** **PPS: 0.32** **Composite score$_1$: 0.67**				
Iteration III (A+B+C)	**APS$_3$: 0.64** **PPS: 0.60** **Composite score$_3$: 1.24**				
Iteration IV (A+B+C+D)	**APS$_2$: 0.80** **PPS: 0.95** **Composite score$_2$: 1.75**				
Iteration V (A+B+C+D+E)	**APS$_2$: 0.80** **PPS: 0.95** **Composite score$_2$: 1.75**				**APS$_1$: 0.15** **PPS= 0.41** **Composite score$_1$: 0.56**

c) *Cost calculation*

Every intervention has a corresponding cost, determined by the intervention type and size of the parcel. Discounts can be taken into account when a cluster reaches a certain size (e.g., a 20% cost reduction for the whole cluster if a cluster reaches a size of 5 ha). Before a parcel is added to the cluster it is confirmed if there is still enough budget left for performing the intervention. If the budget constraint is exceeded when the parcel would be added to the cluster, the cluster growing is stopped and the remaining budget is transferred to the next year's budget.

3) *End of growth*

The end of growth phase is reached when one of the two constraints is not met.

a) *Similarity threshold*

The similarity threshold determines the variability of parcels that is allowed within the cluster. By lowering the threshold, only parcels with a similar composite score will be allowed to enter the cluster, resulting in a more homogenous cluster. As a consequence, the growth of clusters is more rapidly stopped, and the clusters tend to remain smaller, possibly not achieving a large enough size to be entitled to a discounted remedial cost. Therefore, the threshold should be chosen according to a tradeoff between the homogeneity of the clusters on the one hand and the ease and cost of implementing the remediation strategy on the other. The reasoning behind the threshold setting is that when the difference in performance between seed and candidate parcels is large, resources will be used for less urgent parcels or for suboptimal intervention. When the similarity threshold is not met, the cluster growing is stopped and a new seed polygon is found for building the next cluster.

b) *Budget constraint*

The budget constraint limits the amount of resources that can be allocated to interventions in each period. The implementation of a budget constraint in the spatial clustering algorithm ensures that cluster growth cannot lead to exceedi,g the budget for the given period. Once the budget is reached, the attributes of the remaining (unclustered) polygons are adapted to reflect their status for the new period. Next, the clustering can be started for the new period.

IV. CASE STUDY

To demonstrate the capabilities of the proposed spatio-temporal clustering model, it is applied to a case study addressing the remediation of contaminated agricultural parcels. The case study deals with a hypothetical deposition of radioactive Cesium-137 on 1257 agricultural parcels situated in the Maarkebeek Valley in Flanders, Belgium. A remediation plan must be designed for a budget of 500 000 euros per year to ensure that all parcels are remediated so that food can be produced in accordance with the legally set contamination limits. In this case study, five possible remedial interventions are considered: potassium fertilizers, shallow ploughing, deep ploughing, skim and burial ploughing and topsoil removal (Table 5).

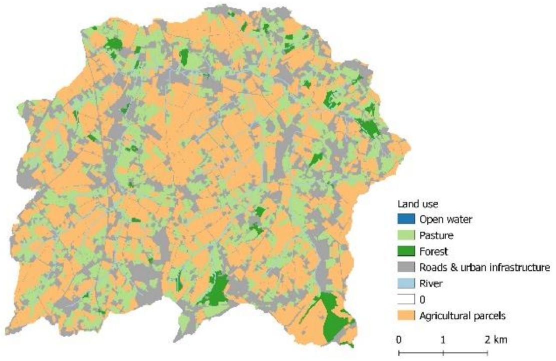


FIGURE 4. LAND USE MAP OF THE MAARKEBEEK WATERSHED IN FLANDERS.

A. *Determination of the Parcel Priority Score*

A parcel is characterized by a set of attributes such as geographic location, environmental characteristics, and agricultural practices. These attributes form the basis for the decision criteria used for determining the PPS (Table 3). The criteria for assessing the priority for remediation of sites with polluted soils were determined from a literature review [28]. Furthermore, each of the criteria was assigned a relative weight based on expert assessment of its importance. The weight is expressed by a linguistic score, which corresponds to a triangular fuzzy number (TFN). TFN are then converted to a quantitative value using the center of gravity method [29].

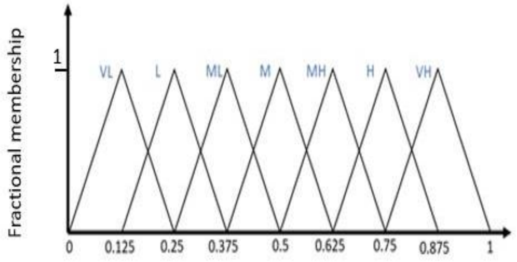


FIGURE 5. MEMBERSHIP FUNCTIONS OF THE LINGUISTIC EXPERT RATINGS USED FOR QUANTIFYING THE CRITERIA WEIGHTS, WITH ABBREVIATIONS VL : VERY LOW, L:LOW, ML: MEDIUM LOW, M: MEDIUM, MH: MEDIUM HIGH, H: HIGH AND VH: VERY HIGH.

The seven criteria and corresponding weights, shown in Table 3, are then used by the CP methodology to determine the feature distance of each parcel to the hypothetical parcel with the highest societal burden and therefore the need for remediation. In Figure 6, the CP methodology, limited to three alternatives and two criteria, is illustrated. The priorities based on this distance for each parcel are shown in Figure 7. Parcels with a low PPS are identified as the most urgent to remediate.

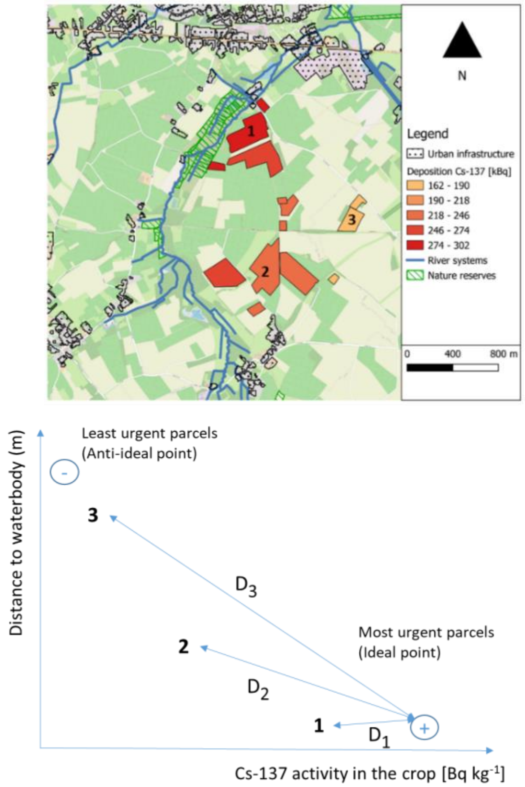


FIGURE 6. REPRESENTATION OF A 2 DIMENSIONAL COMPROMISE PROGRAMMING DISTANCE FOR 3 PARCELS (BOTTOM) AND ITS GEOGRAPHIC REPRESENTATION (TOP).

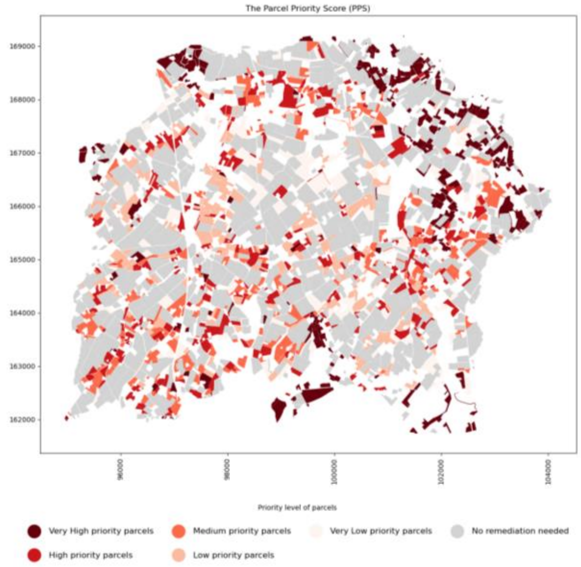


FIGURE 7. PARCEL PRIORITY SCORE (PPS) FOR THE AFFECTED AGRICULTURAL PARCELS, THE LOWER THE PPS THE MORE URGENT THE REMEDIATION.

TABLE 3. CRITERIA USED TO DETERMINE THE PARCEL PRIORITY SCORES (PPS), WITH THE CORRESPONDING WEIGHTS DETERMINED BY EXPERTS

Criterion	Description	Weight
Activity in the food products	The activity of Cs-137 found in the crop after harvest from this field [Bq/kg]	VH
Importance of the food in the local diet	The amount consumed of this product on yearly basis [kg/year]	M
Distance to the urban infrastructure	Distance to the closest urban infrastructure (houses and gardens) [meter]	H
Distance to nature reserves	Distance to the closest nature reserve [meter]	L
Distance to surface water	Distance to the closest surface water (lake/river) [meter]	M
Population density	Population density of the municipality [pp/km2]	H
Erodibility of the parcel	The erosion sensitivity of the field [scale (0 : None - 0.5 : medium - 1: very high)]	L

B. *Determination of the Action Prority Score*

For the determination of the remedial intervention among the five potential remedial actions, six criteria have been selected (Table 4). The applicability of the intervention depends on the parcel's contamination level and the crop type, because some remedial actions are unsuitable for specific agricultural crops or inadequate to reduce the contamination levels below the legal permissible levels. For example, ploughing actions are unfeasible for parcels with perennial crops. The criteria to assess remedial actions can vary largely based on the geographical region, contamination type, stakeholders, and data availability [28].

TABLE 4. CRITERIA USED TO DETERMINE THE ACTION PRIORITY SCORE (APS) OF EACH REMEDIAL INTERVENTION, THE WEIGHTS ARE BASED ON EXPERT JUDGMENT.

Criteria	Description	Weight
Feasibility	The probability that the remediation strategy is implemented successfully.	MH
Incremental Dose	Exposure dose to the workers that need to implement the remediation technique.	MH
Environmental Impact	Risk or actual impact on the living and or non-living environment due to the remediation.	M
Local Impact	Changes to the landscape/ way of life of the population.	MH
The cost of remediation	The total implementation cost of remediation minus the otherwise paid compensation to the farmer. The full remediation cycle is included from investigation to monitoring and waste treatment. [€/ha]	H
Reduction Effectivity	Reduction in activity of agricultural product (compared to doing nothing). [%]	VH

The five remedial alternatives are scored on the six criteria that produce the alternative-criterion matrix (Table 5), which is the basis for the distance calculations by the CP. More information on the determination of the criteria scores in the alternative-criterion matrix can be found in [30].

TABLE 5. ALTERNATIVE-CRITERION MATRIX FOR THE FIVE REMEDIAL ALTERNATIVES, SCORING THEM ON SIX CRITERIA.

	Feasibility	Incremental dose	Environmental impact	Local impact	Direct cost of application	Reduction Effectivity
Potassium fertilizers	H	L	H	L	66 (Yearly)	69
Shallow ploughing	VH	MH	VL	L	39 (Single time)	50
Deep ploughing	M	H	M	MH	53 (Single time)	70
Skim and burial ploughing	L	H	M	MH	95 (Single time)	87.5
Topsoil removal	L	H	VH	VH	24490 (Single time)	93.5

The incorporation of the temporal dynamics in this case study is necessary since the values of certain decision variables change through time. Because of natural attenuation, which causes the mass, toxicity, volume or concentration of contaminants in the soil or groundwater to reduce over time. This implies that the contamination decreases over time without the interference of specific remedial actions. For radioactive contaminations the reduction of the contaminant is strongly determined by the radioactive decay, the radionuclide's half-live. For a remedial action to be considered feasible, it should be able to reduce the contamination levels below the legally allowed limits. From the dynamic nature of the contamination, it follows that, after a certain period of time other remedial options can become more effective and outperform the previously selected option. Consequently, the remedial actions for each parcel should be revised to ensure they are still optimal for this time period. For this case study, the weights are not considered to change between periods.

C. Individual per parcel solution

For each individual parcel and for each time period, an APS score for each feasible remediation technique can be calculated. This is illustrated in Figure 8 for a cereal parcel. For this specific field, only four remedial actions are feasible, and deep ploughing is considered the most optimal since it has the lowest value. Topsoil removal is the second-most optimal remedial technology.

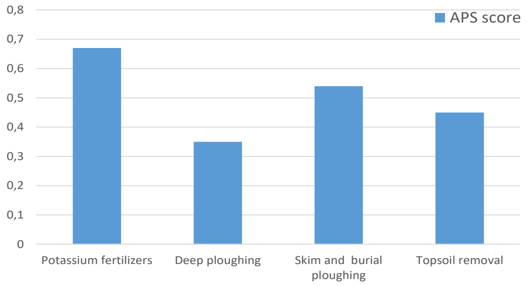


FIGURE 8. ACTION PRIORITY SCORE (APS) FOR THE DIFFERENT CANDIDATE REMEDIAL ACTIONS ON AN AGRICULTURAL PARCEL WITH CEREAL CULTIVATION.

In Figure 9, the optimal remediation technique for each parcel, based on the technique with the lowest APS, is shown.

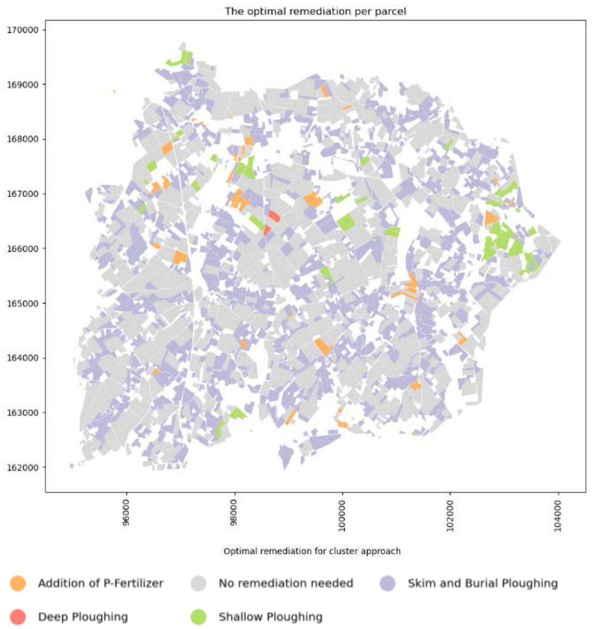


FIGURE 9. PROPOSED REMEDIATION PLAN BASED ON THE OPTIMAL REMEDIAL ACTION FOR EACH PARCEL.

D. Spatio-temporal cluster solution for the affected region

With the spatio-temporal cluster approach, a multi-period action plan can be designed, taking into account when and how to remediate the parcels. For the same area, the model proposes a remedial technique and timing. Both can be found in Figures 10 and 11, respectively.

The difference in remedial technologies between Figures 9 and 10 can be explained by the clustering of parcels and the changing of some of the parcel characteristics due to the delayed remediation. The remedial action "food restriction" found in Figure 10 is for agricultural parcels where, due to the physical decay process described above, the food crops can be produced with radioactivity below the permissible levels without the need for a remedial action given the time elapsed since the deposition of the radionuclides. It is clear that the model will seek optimal homogenous clusters, where the solution is optimal overall and not for each individual parcel.

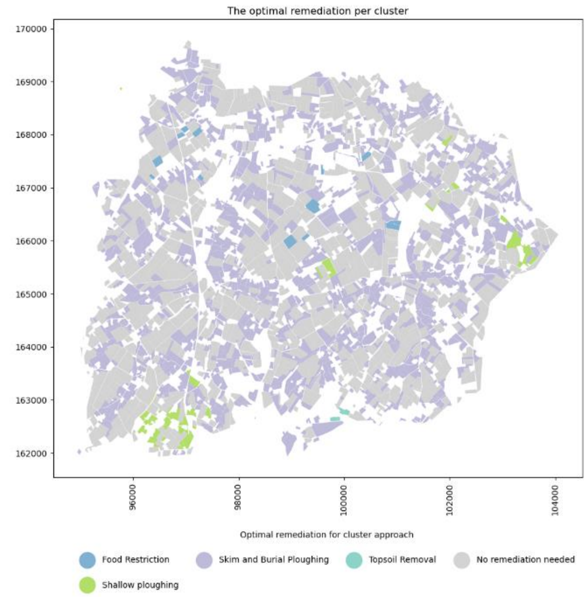


FIGURE 10: THE REMEDIAL TECHNOLOGIES PROPOSED BY THE SPATIO-TEMPORAL CLUSTERING ALGORITHM WITH A SIMILARITY THRESHOLD OF 0.025.

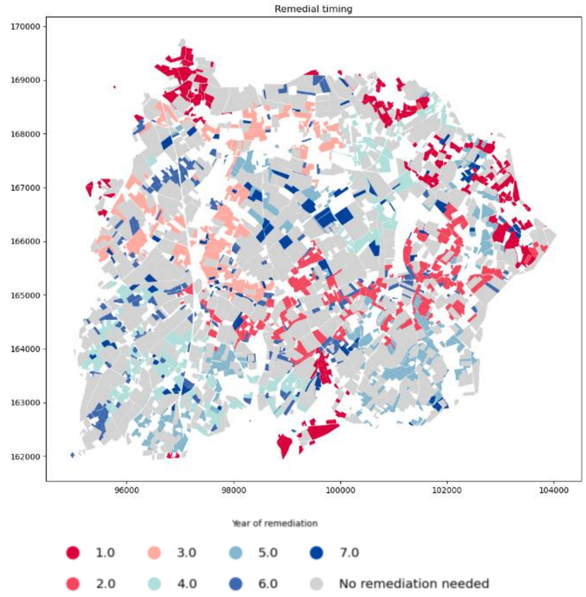


FIGURE 11. THE TIMING OF REMEDIATION PROPOSED BY THE SPATIO-TEMPORAL CLUSTERING ALGORITHM WHEN THE SIMILARITY THRESHOLD IS SET TO 0.025.

E. Intracluster variability

The variability of the PPS score within the cluster should be as low as possible to make sure that resources are used for the most urgent parcels. When the similarity threshold is set to 0, as in Figure 12, the clusters consists of only the seed parcel. It is clear that as cluster rank increases, so does the value of the cluster's PPS score, demonstrating the prioritization of resources for the most important parcels.

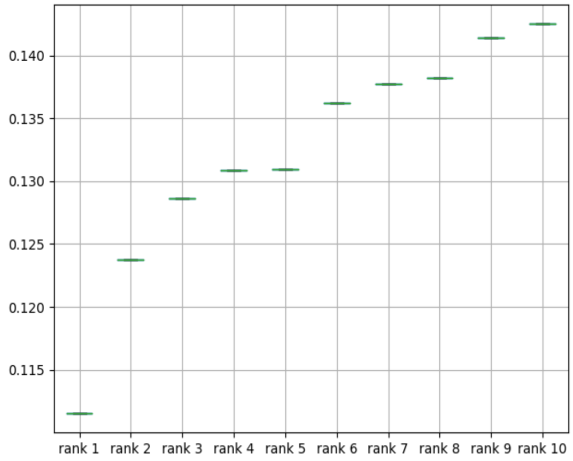


FIGURE 12. PPS SCORE FOR THE 10 HIGHEST RANKED CLUSTERS, WHICH ARE EQUAL TO THE 10 HIGHEST RANKED SEED PARCELS FOR A SIMILARITY THRESHOLD OF 0.

With an increasing similarity threshold (see Figures 13 and 14), the variability of the PPS within the cluster is allowed to increase. Furthermore, it is important to observe the increased presence of outliers due to the higher similarity threshold. This can be important when the seed parcel is the outlier, because then resources are potentially used on a less important site first.

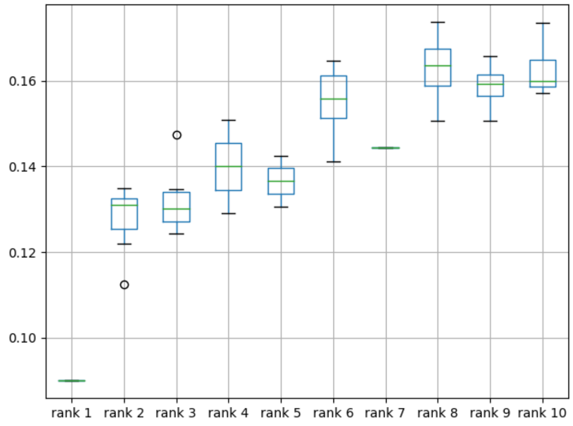


FIGURE 13. BOXPLOTS REPRESENTING THE VARIABILITY OF THE PPS SCORE WITHIN THE 10 HIGHEST RANKED CLUSTERS, WHEN THE SIMILARITY THRESHOLD IS 0.025.

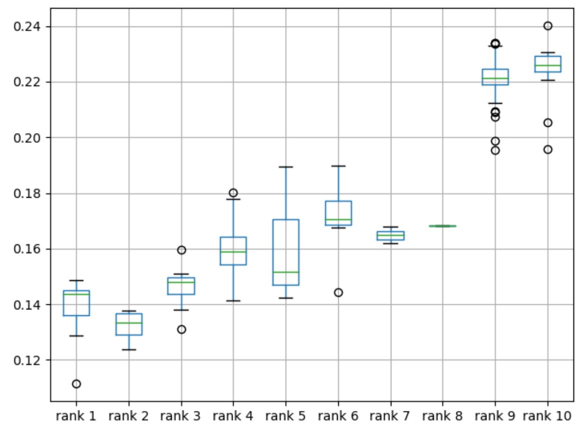


FIGURE 14. BOXPLOTS REPRESENTING THE VARIABILITY OF THE PPS SCORE WITHIN THE 10 HIGHEST RANKED CLUSTERS, WHEN THE SIMILARITY THRESHOLD IS 0.05.

It is clear that a higher similarity threshold results in more resources going to less important parcels, but on the other hand, it results in larger clusters and therefore lower operational costs. When lowering the similarity threshold for more optimal decision making, the overall cost of remediation will increase, resulting in more time needed for the remediation of the affected region. This effect can be seen in Figure 15, where the remediation will take nine years instead of seven, increasing the budget by around 1 million euros.

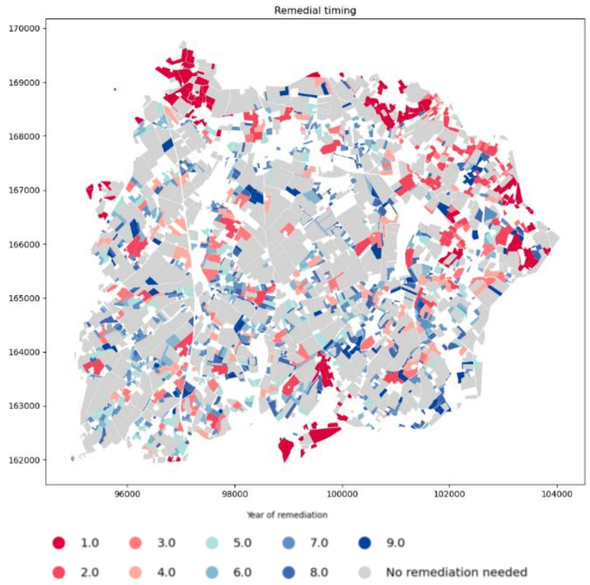


FIGURE 15. THE TIMING OF REMEDIATION PROPOSED BY THE SPATIO-TEMPORAL CLUSTERING ALGORITHM WHEN THE SIMILARITY THRESHOLD IS SET TO 0.01.

V. DISCUSSION

This case study shows the complexity of designing spatio-temporal remedial schemes. Therefore, the use of a GIS-MADM based DSS, as proposed in this paper, could help decision makers find clarity and see the impacts of certain decisions. A major benefit of these tools is the ability to do scenario analysis and uncertainty analysis. The impact of varying degrees of uncertainty in this decision context is described in [31]. Further the use of these dynamic MADM approaches allows for a shift to a more adaptive management paradigm [32].

The spatio-temporal MADM relies heavily on the PPS and APS of a parcel; therefore, the determination of these scores should be done with great care. The determination of the specific applicable criteria and weights is not only the work of experts, but it is highly suggested to take into account all stakeholders to ensure a solution supported by society is proposed [28].

For the purpose of this research, CP was used with a Euclidean distance measure, but other distance metrics are possible (e.g., Manhattan distance). Because of the use of two distance-based metrics with similar range, the composite distance score still has a physical meaning (distance to the ideal or anti-ideal situation).

Figures 12 to 14 show the effect of the increasing similarity threshold on the variability of PSS scores within clusters. A larger similarity threshold allows more variation within the cluster; therefore, less optimal clusters are formed and more deviation from the optimal per-parcel-solution is allowed. However, larger clusters give rise to lower operational costs, resulting in cheaper and faster remediation. Decision makers can decide what is the best setting for their own specific case, but a rule of thumb to determine the initial similarity threshold is half of the range of the APS values. The budget constraint limits the amount of interventions per year, therefore, a lower budget will spread the remediation over more years. This increase in remediation time could potentially change the remedial actions for parcels because of delayed remediation.

The reduced cost of remediation for larger units is the main driver for the introduction of remedial management clusters. For this specific case study, expert-based estimations for the discounts were used because empirical data for these large-scale remedial actions is not widely available. Nevertheless, they should be determined with great care and potentially adapted during the remedial process to improve the model estimations.

The proposed technical implementation of the budget constraint stops the remediation if the most optimal neighbor of the cluster with the specific remedial action exceeds the available budget, whereby the remaining budget is transferred to the next year. This transfer has a low impact, when the yearly budget exceeds largely the remediation cost of a single cluster.

When working with polygon-based datasets, topological errors, such as gaps, may occur. Relying solely on these topological relationships can have major impacts on determining the neighbours. Our EMT approach is less impacted by these errors.

Other cases could benefit from a similar approach. For example, when afforesting a large region, not all sites can be afforested at the same time because it is a very costly and labor intensive intervention. Additional, every plot has a certain suitability and urgency to be afforested. In addition, afforesting connected parcels with a similar tree composition would severely reduce the cost of planting and also improve the ecological connectivity of the landscape. Therefore, finding optimal clusters of parcels to be afforested with similar tree compositions could be facilitated with our approach. A similar approach for raster datasets was already reported by [14].

VI. Conclusions and future work

With the proposed spatio-temporal clustering approach, dispersed polygons can be clustered in space and time and be assigned the most optimal intervention type under a budget constraint. This allows decision makers to form multi-period remedial schemes to address the environmental disaster. The approach also gives decision makers the possibility to do scenario analysis and uncertainty analysis to better understand the impact of the different parameters in the model. In addition, the approach shows promise for other fields of application. More research on the impact of the similarity threshold is needed. In addition, the introduction of off-site impacts (e.g., transport and re-deposition of contaminated sediment) should be incorporated in the MADM criteria [33] to better mimic the contamination behavior. Future research should consider multiple consecutive remedial actions rather than single ones [20], to be more in line with the reality of remediation.

Acknowledgment

This work was supported by a PhD grant for Floris Abrams from the Belgian Nuclear Research Centre (SCK CEN).

References

[1] F. Abrams, L. Sweeck, J. Camps, D. Cattrysse, and J. Van Orshoven, “Spatio-Temporal Clustering of Polygon Objects and per Object Interventions Optimizing Remediation of Spatially Dispersed Contaminated Parcels Under an Annual Budget Constraint,” in *The Fourteenth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2022)*, 2022, pp. 1–6.

[2] U.S. EPA, “Superfund Optimization Progress Report,” 2020.

[3] J. Malczewski and C. Rinner, *Multicriteria Decision Analysis in Geographic Information Science*. 2015.

[4] R. Anderson, J. Norrman, P. E. Back, T. Söderqvist, and L. Rosén, “What’s the point? The contribution of a sustainability view in contaminated site remediation,” *Sci. Total Environ.*, vol. 630, pp. 103–116, 2018.

[5] J. Malczewski, *GIS and Multicriteria Decision Analysis*. 1999.

[6] C. D. Gamper and C. Turcanu, “Multi-criteria analysis: A tool for going beyond monetization?,” *Tools Policy Formul. Actors, Capacit. Venues Eff.*, pp. 121–141, 2015.

[7] J. Kingscott and R. J. Weisman, “Cost Evaluation for Selected Remediation Technologies,” *Remediat. J.*, vol. 12, no. 2, pp. 99–116, Mar. 2002.

[8] S. Polat, A. Aksoy, and K. Unlu, “A Fuzzy Rule Based Remedial Priority Ranking System for Contaminated Sites,” *GROUNDWATER*, vol. 53, no. 2, pp. 317–327, 2015.

[9] K. Zhang, G. Achari, and Y. Pei, “Incorporating linguistic, probabilistic, and possibilistic information in a risk-based approach for ranking contaminated sites,” *Integr. Environ. Assess. Manag.*, vol. 6, no. 4, pp. 711–724, 2010.

[10] Y. Lin, J. Hoover, D. Beene, E. Erdei, and Z. Liu, “Environmental risk mapping of potential abandoned uranium mine contamination on the Navajo Nation, USA, using a GIS-based multi-criteria decision analysis approach,” *Environ. Sci. Pollut. Res.*, vol. 27, no. 24, pp. 30542–30557, 2020.

[11] M. A. B. Promentilla, T. Furuichi, K. Ishii, and N. Tanikawa, “Evaluation of remedial countermeasures using the analytic network process,” *Waste Manag.*, vol. 26, no. 12, pp. 1410–1421, 2006.

[12] I. Linkov, A. Varghese, S. Jamil, T. P. Seager, G. Kiker, and T. Bridges, “Multi-criteria decision analysis: A framework for structuring remedial decisions at contaminated sites,” in *Comparative Risk Assessment and Environmental Decision Making*, 2004, vol. 38, pp. 15–54.

[13] L. Rosen *et al.*, “SCORE: A novel multi-criteria decision analysis approach to assessing the sustainability of contaminated land remediation,” *Sci. Total Environ.*, vol. 511, pp. 621–638, Apr. 2015.

[14] P. Vanegas, D. Cattrysse, A. Wijffels, and J. Van Orshoven, “Finding sites meeting compactness and on- and off-site suitability criteria in raster maps,” in *2nd International Conference on Advanced Geographic Information Systems, Applications, and Services, GEOProcessing 2010*, 2010, pp. 15–20.

[15] M. Fleischmann, “momepy: Urban Morphology Measuring Toolkit,” *J. Open Source Softw.*, vol. 4, no. 43, p. 1807, 2019.

[16] Z. Xu, “On multi-period multi-attribute decision making,” *Knowledge-Based Syst.*, vol. 21, no. 2, pp. 164–171, 2008.

[17] Y. Chen and B. Li, “Dynamic multi-attribute decision making model based on triangular intuitionistic fuzzy numbers,” *Sci. Iran.*, vol. 18, no. 2 B, pp. 268–274, 2011.

[18] M. Karatas, “Multiattribute Decision Making Using

Multiperiod Probabilistic Weighted Fuzzy Axiomatic Design," *Syst. Eng.*, vol. 20, no. 4, pp. 318–334, 2017.

[19] Q. Dong and Y. Guo, "Multiperiod multiattribute decision-making method based on trend incentive coefficient," *Int. Trans. Oper. Res.*, vol. 20, no. 1, pp. 141–152, 2013.

[20] R. Estrella, D. Cattrysse, and J. Van Orshoven, "An Integer Programming Model to Determine Land Use Trajectories for Optimizing Regionally Integrated Ecosystem Services Delivery," pp. 1–26, 2016.

[21] J. Tian, Z. Huo, F. Ma, X. Gao, and Y. Wu, "Application and Selection of Remediation Technology for OCPs-Contaminated Sites by Decision-Making Methods," *Int. J. Environ. Res. Public Health*, vol. 16, no. 11, Jun. 2019.

[22] C. A. Salt and M. C. Dunsmore, "Development of a spatial decision support system for post-emergency management of radioactively contaminated land," *J. Environ. Manage.*, vol. 58, no. 3, pp. 169–178, 2000.

[23] A. L. Yang, G. H. Huang, X. S. Qin, and Y. R. Fan, "Evaluation of remedial options for a benzene-contaminated site through a simulation-based fuzzy-MCDA approach," *J. Hazard. Mater.*, vol. 213, pp. 421–433, Apr. 2012.

[24] L. Bai *et al.*, "TOPSIS-Based Screening Method of Soil Remediation Technology for Contaminated Sites and its Application," *Soil Sediment Contam.*, vol. 24, no. 4, pp. 386–397, 2015.

[25] V. Belton and T. J. Stewart, *Multiple criteria decision analysis: An integrated approach*. 2002.

[26] N. H. Zardari, A. Kamal, S. Sharif Monirussaman, and Y. bin Zulkifli, *Weighting Methods and their Effects on Multi-Criteria Decision Making Model Outcomes in Water Resources Management*. 2015.

[27] M. Fleischmann, A. Feliciotti, O. Romice, and S. Porta, "Morphological tessellation as a way of partitioning space: Improving consistency in urban morphology at the plot scale," *Comput. Environ. Urban Syst.*, vol. 80, no. May 2019, p. 101441, 2020.

[28] F. Abrams, L. Sweeck, L. Hendrickx, C. Turcanuc, R. Estrella, and J. Van Orshoven, "Multi-criteria Decision Analysis to Support the Remediation of Polluted Soils: a Review," *J. Environ. Plan. Manag.*, 2022.

[29] W. J. Wang and L. Luoh, "Simple computation for the defuzzifications of center of sum and center of gravity," *J. Intell. Fuzzy Syst.*, 2000.

[30] L. Hendrickx, F. Abrams, L. Sweeck, and J. Van Orshoven, "Supporting Remediation of Agricultural Land Contaminated With Cesium-137: A Multi-criteria Decision Workflow Accounting for Uncertainties," KULeuven, 2021.

[31] F. Abrams, L. Hendrickx, L. Sweeck, J. Camps, D. Cattrysse, and J. Van Orshoven, "Accounting for Uncertainty and Disagreement in Multi-criteria Decision Making Using Triangular Fuzzy Numbers and Monte Carlo Simulation: A Case Study About Selecting Measures for Remediation of Agricultural Land After Radioactive Contamination," in *Real Life Applications of Multiple Criteria Decision Making Techniques in Fuzzy Domain*, Springer Nature, 2022.

[32] I. Linkov, F. K. Satterstrom, G. Kiker, C. Batchelor, T. Bridges, and E. Ferguson, "From comparative risk assessment to multi-criteria decision analysis and adaptive management: Recent developments and applications," *Environ. Int.*, vol. 32, no. 8, pp. 1072–1093, 2006.

[33] R. Estrella, P. Vanegas, D. Cattrysse, and J. Van Orshoven, "Trading off Accuracy and Computational Efficiency of an Afforestation Site Location Method for Minimizing Sediment Yield in a River Catchment," 2014, no. c, pp. 94–100.

Green Storage: Parallel File Systems on ARM

Timm Leon Erxleben[*], Kira Duwe[*], Jens Saak[†], Martin Köhler[†] and Michael Kuhn[*]
[*]Otto von Guericke University Magdeburg
Magdeburg, Germany
E-mail: timm.erxleben@ovgu.de, kira.duwe@ovgu.de, michael.kuhn@ovgu.de
[†]Max Planck Institute for Dynamics of Complex Technical Systems
Magdeburg, Germany
E-mail: saak@mpi-magdeburg.mpg.de, koehlerm@mpi-magdeburg.mpg.de

***Abstract*—Parallel distributed file systems are typically run on dedicated storage servers that clients connect to via the network. Regular x86 servers provide high computational power, often not required for storage management and handling I/O requests. Therefore, storage servers often use low core counts but still have a relatively high idle power consumption. This leads to high energy consumption, even for mostly idle file systems. Advanced Reduced Instruction Set Computer Machines (ARM) systems are very energy-efficient but still provide adequate performance for file system use cases. Leveraging this fact, we built an ARM-based storage system, on which we tested different parallel distributed file systems. We compare the performance and energy efficiency of x86 and ARM systems using several metrics. Analysis of the different file systems on the ARM system shows that energy efficiency highly depends on the architecture and the used file system. Results show that while our ARM-based approach currently provides less throughput per Watt for reads, it achieves an approximately 174 % higher write efficiency when compared to a traditional x86 Ceph cluster.**

Keywords*—*energy efficiency; parallel distributed file systems; x86; ARM

I. Introduction

Storage systems are scaled up steadily to satisfy increasing storage demands, leading to growing energy consumption [2]. High-Performance Computing (HPC) storage systems are currently built from regular x86 servers, whose computing power is not fully utilized by storage applications. Traditional x86 servers feature a relatively high power consumption even when idle: It is not uncommon to measure idle consumption of more than 100 W for just the processor, main memory, and mainboard. In comparison, low-power ARM computers are often required to stay below 5–10 W maximum consumption by design. To offset the high idle consumption of x86 servers, they have to be equipped with large amounts of storage devices, such as hard disk drives (HDDs) and solid-state disks (SSDs). However, depending on the used network interconnect, only a limited number of devices can be saturated. For instance, on a 100 Gbit/s network, two to three Non-Volatile Memory Express (NVMe) SSDs are enough to provide the necessary throughput and more devices cannot be used to their full extent. This proportion gets even worse on slower networks.

Therefore, we evaluate the use of low-energy ARM-based single-board computers (SBCs) as a replacement for traditional servers in storage systems. To assess the feasibility of an ARM-based storage system, we evaluated the ARM-based cluster using CephFS, OrangeFS, MooseFS and GlusterFS. We compared the performance and energy efficiency of the different configurations to an OrangeFS test cluster at the University of Hamburg, using different metrics and workloads. Furthermore, we compared it to a productive CephFS cluster running at the computer science faculty of the Otto von Guericke University Magdeburg, to validate the approach against modern hardware.

The contributions of our paper are:

1) We propose to apply the energy-delay product, typically used to evaluate the energy efficiency of computations, as a metric for storage systems as well to measure energy efficiency while still accounting for the performance needed by HPC applications.
2) We show that low-power ARM-based storage clusters can achieve throughput efficiencies comparable to, or even exceeding, traditional x86 systems.

This paper is based on a previous conference paper [1]. Since then, we analyzed and tested two additional file systems, MooseFS and GlusterFS, and interpreted the measured power consumption with respect to performance and energy efficiency. We also included one additional cluster with different hardware characteristics in our evaluations.

The remainder of the paper is organized as follows. In Section II, the used file systems are briefly described followed by a summary of related works in Section III. Section IV describes the benchmarks which were done and discusses metrics that can be derived from the measurement data. Next, in Section V all cluster setups, ARM and x86, are described, followed by the presentation of the results. Results and setups are discussed in Section VI. Section VII concludes the paper.

II. Background

This section introduces background on used technologies, such as the used parallel file systems. The information is taken from the respective file system's documentation if not referenced otherwise.

A. Ceph

Ceph [3] is a popular, clustered object store, which is highly scalable due to its Controlled Replication Under Scalable Hashing (CRUSH) placement algorithm, which enables all participating services, that can access the cluster map to locate and place objects [4]. A typical Ceph cluster is made of Object

Storage Devices (OSDs), monitoring and management services. All components may be redundant to enable automatic failover.

Apart from access through the library `librados`, many interfaces might be used. The POSIX access via CephFS, realized by additional Metadata Services (MDSs) interacting with Ceph storage pools, is particularly interesting for HPC systems. This is because POSIX is often used by scientific high-level I/O libraries like HDF5 or NetCDF and ensures portability of many applications. Even so, its semantics are mostly too strict for typical HPC I/O requirements and can impair performance [5]. CephFS has a rich feature set, including replication, multiple storage pools, file systems, snapshots, and high control over data placement.

B. OrangeFS

OrangeFS is a traditional parallel distributed file system designed for HPC [6][7]. Only one type of server is needed, which can handle both data and metadata, though it can be configured to handle only one type.

In OrangeFS, data is striped according to a distribution function that can be specified for each file. The default is to start at a random server and use all servers in a round-robin fashion with a stripe size of 64 KiB. Unlike Ceph, which uses its own object store *Bluestore* [8], OrangeFS relies on a separate local file system.

As of the current version, 2.9.8, there are no redundancy features for data that is not marked as read-only, though this is planned for OrangeFS version 3 [9]. Many interfaces may be used to interact with OrangeFS. Most popular choices include access via the OrangeFS Linux kernel module or direct access using the library `libpvfs2`. Noteworthy is the direct Message Passing Interface I/O (MPI-IO) support by using ROMIO's [10] Abstract-Device Interface for I/O (ADIO), for which OrangeFS provides an implementation [11].

C. MooseFS

MooseFS [12] is a POSIX-compliant parallel distributed file system designed for Big Data applications [13]. It makes use of different server types for metadata and data storage, monitoring and metadata backups.

Metadata is managed using the so-called master server for any accesses and several metalogger servers for backup purposes. Unlike in the other used parallel file systems, metadata is completely held in memory. Persistency is guaranteed by periodic on-disk backups and an on-disk journal for modifying operations.

Data is striped with a hard-coded size of 64 MiB and distributed to the chunk servers, which provide persistent storage using the underlying local file system. The distribution is random but prioritizes chunk servers with a lower load [14]. For data safety, each file has a replication goal.

D. GlusterFS

GlusterFS [15] is a parallel distributed file system for cloud storage and media streaming. Like OrangeFS, GlusterFS has only one type of server. All GlusterFS severs form a Trusted Storage Pool (TSP), which provides attached storage, called bricks, for volumes. Each volume creates its own namespace that clients can mount. Multiple volumes of different types may be created on top of a TSP.

The different volume types specify the distribution of data in the cluster. Distributed and replicated volumes distribute files without striping and are therefore not suitable for HPC applications. However, dispersed volumes make use of striping and provide data safety via redundant data blocks using erasure coding. This type of volume provides the parallel access needed to satisfy the performance requirements of parallel applications. The block size depends on the number of storage servers and the ratio of redundant blocks.

In GlusterFS, no separate metadata handling is needed because all participants can determine file positions by hashing. Unix file metadata, like access time or permissions, is stored in the inodes of the underlying file system. Other GlusterFS-specific metadata is stored using extended file attributes. In dispersed volumes, the metadata is duplicated to each file fragment, which contains all blocks of that file on this server. These file fragments are stored as a regular file on the servers.

Though GlusterFS is not specifically designed for HPC applications the use of erasure coding via dispersed volumes was interesting for the comparison with the other file systems.

III. State of the Art and Related Work

There have been various endeavors to measure and increase the energy efficiency of large systems, as energy consumption is becoming a possible constraint on HPC systems in the future. Many different aspects have to be considered, ranging from the system's energy efficiency to the scalability of the applications. As ARM processors aim to offer better energy efficiency, they have been heavily studied across the years [16–18]. Deployments, such as Fugaku [19], show that they can provide competitive performance and even work in exascale systems. Earlier research on systems like Tibidabo at Barcelona Supercomputing Center indicated that single instruction, multiple data stream (SIMD) instructions limited to single precision were a severe bottleneck for the performance [17][18][20].

Energy efficiency is also a relevant aspect in distributed systems, as examined for peer-to-peer systems. A survey by Brienza et al. [21] showed that often simple energy models were used, disregarding other hardware components like intermediate routers. An early approach, and still very prominent solution to energy savings in storage, is sending idle peers to sleep [22]. However, it introduces problems when the load varies. To have systems benefit from the increased energy efficiency, in the long run, applications have to be considered as well. The optimization towards energy efficiency comes indeed with its challenges for applications [20][23–25]. Reducing the performance of a single core, in order to cap the power consumption, means that scalability is of increased importance [20].

Gudu and Hardt evaluated the use of an ARM-based Ceph cluster, made of Cubieboards, as a replacement for traditional network-attached storage (NAS) controllers [26]. They measured the throughput of their cluster via Ceph's Reliable Autonomic Distributed Object store (RADOS) and RADOS Block Device (RBD) access and found that the Cubieboard cluster is a viable alternative to NAS controllers. However, the limited network capabilities were the bottleneck of the system.

Apart from using low-power hardware [27], there have been efforts to reduce the power consumption of existing HPC storage clusters [28][29]. For example, it was proposed to assign subsets of storage clusters to individual users and only run a specific subset at full power when an assigned user uses the compute-cluster [30].

Considering that local file systems are often part of the storage stack, their influence on energy efficiency and performance were analyzed, in [31], using simulated workloads of web, database, and file servers. It was found that the choice of file system and its configuration greatly influence performance and energy efficiency. However, no file system performed best for all workloads.

In contrast to Gudu and Hardt, we measure data throughput at the CephFS level and evaluate ARM-based clusters as a replacement for HPC storage clusters.

IV. Benchmark and Metrics

We measured the performance of the clusters for data-throughput oriented workloads. The benchmark comprised sequential, independent accesses from one to four clients using IOR v3.3 [32] with the POSIX backend, individual files per client and five iterations for each data point. The transfer size was set to 4 MiB, which corresponds to the default stripe size of CephFS and is aligned to the stripe size of the other parallel file systems. On the x86-based Ceph cluster, 96 GiB were written and read. The amount of data was reduced to 36 GiB for the other two clusters to keep run-times manageable.

For every iteration of the measurements, the power consumption of the storage cluster was measured using the methods as described in Section V. As a result, several energy efficiency metrics can be derived from the collected data. However, choosing a specific metric is not trivial, as there is no single optimal metric indicating energy efficiency [33].

We decided to compare the results obtained by using the **energy-delay product** (EDP) [34], **throughput per Watt** and **capacity per Watt** [35].

Throughput per Watt is a commonly used metric for evaluating and comparing storage energy efficiency. The transferred data may differ between systems, so it is well suited to compare systems that greatly vary in their performance. However, this metric alone is insufficient when analyzing and optimizing storage systems, as no insight into performance is given. Geveler et al. [23] found that for simulations, in some cases, energy savings might lead to performance drops. In such cases, they motivated using the EDP as a fused metric describing energy efficiency and performance at once. The EDP is computed as the product of the total energy E consumed while performing a task and the time t needed to complete the task (Equation (1)). Depending on the performance requirements, the time may be weighted [36]. As we want to focus on energy consumption, we set $w = 1$.

$$\text{EDP} = E \cdot t^w, \quad w \in \mathbb{N} \tag{1}$$

Though the energy-delay product was initially developed for hardware design, it is also useful when evaluating software, as done by Georgiou et al. [37]. Nevertheless, the amount of work needs to stay constant to compare different systems, so only the two ARM setups are compared using the EDP. Because its unit is hard to interpret and even changes with different weights, we normalized the EDP using the lowest value per comparison.

The third metric considered measures the capacity of the storage system per Watt. Because of growing storage demands and, therefore, growing storage systems, optimizing systems regarding this metric is critical for the cost-efficient and environmentally friendly operation of data centers.

V. Evaluation

In this section, the hardware and software setup is described, followed by an analysis of the respective clusters' theoretical peak performance and the presentation of the results.

A. Reference Cluster 1

The first reference cluster is a five node subset of a research cluster at the University of Hamburg. Each node has two Intel Xeon X5650 CPUs, each featuring six cores at 2.67 GHz, 11 GB RAM, and two Intel 82574L Gigabit Network Interface Cards (NICs). One node is equipped with a 250 GB Western Digital WD2502ABYS HDD [38], while the other nodes are equipped with a 250 GB Seagate ST3250318AS HDD [39]. A ZES Zimmer LMG 450 power meter was used to measure the power consumption of this setup. The five nodes consumed **460.21 W** on average in idle state with a standard deviation of 18.43 W, measured over one hour, with HDDs spun up. The clients used to benchmark this reference cluster were four servers of the same specification.

We used OrangeFS version 2.9.8 as file system for its straightforward setup and good comparability to the ARM-cluster. One node was used exclusively for metadata storage, while the other four nodes provided data storage. The used block size was the default of 64 KiB. The same configuration of OrangeFS was later used on the ARM-based cluster.

B. Reference Cluster 2

The second reference cluster is a four-node subset of the productive Ceph cluster running at the computer science faculty at the Otto von Guericke University using Ceph 16.2.7 deployed as containers. Three nodes of the subset are part of the Supermicro AS 2124BT-HNTR [40] multi-node system, each of which is equipped with four Intel P4510 NVMe SSDs [41]. The fourth server is a Gigabyte R282-Z94 [42] equipped with one Intel P4510 NVMe SSD and eight Samsung

MZQL23T8HCJS-00A07 NVMe SSDs [43]. All nodes are connected by 100 Gbit Ethernet, with a separate 100 Gbit network for communication between Ceph OSDs. Though Ceph does not exclusively use the nodes, they are idle most of the time. The average idle power consumption of the four nodes was measured to be **699.3 W**. This power measurement was done on a Sunday since the servers are mostly idle on the weekend. It lasted for one hour, starting at 14:00, and had a standard deviation of 13.98 W. While running, the benchmark power consumption peaked at 1,057 W. The existing monitoring solution, gathering power samples over IPMI every 15 seconds, was used to collect power samples.

For each SSD, two Ceph OSDs are deployed. The Ceph monitor and a standby metadata service are located at the Gigabyte server, while the active metadata service runs on one of the Supermicro servers. Ceph pools use the default replication settings and, therefore, produce three replicas of the data and return to the client after two replicas are written. The clients used for the benchmark were four servers equipped with an AMD Epyc 7443, with 24 cores at 2.85 GHz, 128 GB RAM, and 100 Gbit Ethernet.

C. ARM Cluster

a) Cluster Setup: The low-power cluster is built of six Odroid HC4 nodes featuring the Amlogic S905X3 SoC, with four cores at 1.8 GHz, 4 GiB DDR4 RAM, two SATA-3 ports, and a 1 Gbit NIC [44] (see Figure 1). We decided to use the Odroid HC4 instead of more typical SBCs like the Raspberry Pi [45] due to its native SATA ports. Four of the nodes, nodes A1–A4, are equipped with two 1 TB WD Black HDDs [46] and one, node C, is equipped with two 512 GB Samsung V-NAND SSD 860 PRO SSDs [47]. One exception was made for GlusterFS, where the SSDs were exchanged with two more HDDs. This decision will be discussed later on. The last remaining node, node B, has no disks and is intended for monitoring purposes. All nodes are connected to a Netgear GS110EMX switch [48].

The ARM-cluster nodes run on Armbian Buster 21.08.8, which uses Linux 5.10.81-meson64. Armbian [49] is a Linux distribution based on Debian, which is modified and optimized for use on SBCs. The pre-installed *Petitboot* was erased from the HC4's flash memory to use the *uboot* bootloader, which is part of the Armbian image.

The complete cluster, including the switch, is powered by an MW HRP450-15 PSU [50] and consumes 56.36 W, measured over one hour with a standard deviation of 0.14 W, in idle state, with HDDs spun up. For comparison with the reference cluster, which does not include the switch in the power measurements, we subtracted the average idle power of the switch, which was measured to be 15.46 W, with a standard deviation of 1.13 W over one hour. The adjusted idle power consumption of the ARM cluster, therefore, is **40.9 W**. The highest peak in power consumption measured while running the benchmark was 63.68 W, which was observed for writes with four clients and MooseFS.

For power measurements, the ZES Zimmer LMG 450 [51] was used to measure the power consumption of the PSU for the whole cluster. The power meter was connected to a BananaPi M1 via USB, which collects samples with 20 Hz.

The clients used to perform the benchmark were four Dell Precision 3650 Tower workstations [52] each with an Intel Core i7-11700 CPU with 8 cores at 2.5 GHz, 8 GB RAM, and a 1 Gbit NIC. They were connected via the network infrastructure of the Max Planck Institute Magdeburg.

We used Debian Bullseye 5.10.46-5 on the clients, which uses Linux 5.10.0-8. OpenMPI 4.1.0 with the included MPI-I/O implementation OMPIO was installed from the Debian Buster Repository for parallel benchmarks. All storage nodes and clients use Network Time Protocol (NTP) to synchronize their clocks with node C.

The network topology is visualized in Figure 1, where dotted lines depict the devices included in the power measurement.

b) Parallel File System Configuration: We compared four different parallel file systems on the ARM cluster: CephFS, OrangeFS, MooseFS and GlusterFS. Though not all of them are originally designed for the same purpose and workloads, they can be configured to perform reasonably well for the parallel coordinated access. The different architectures and features of the file systems were useful to determine the capabilities of the ARM-based cluster.

We used Ceph version 14.2.21, which is available in the Buster backports repository. One OSD was deployed for each storage device. Node C, which was equipped with SSDs, additionally ran one MDS. The Ceph monitor and management daemon ran on node B, which has no disks attached. The two storage pools needed for CephFS used different CRUSH rules to distribute objects. While the data pool used all HDDs and managed replicas on the node level, the metadata pool used the two SSDs and managed replicas on the OSD level. Both pools were configured to use 64 placement groups, to produce two replicas and to return immediately after one replica is written. The relaxed replication settings allowed a fairer comparison with the other file systems.

We built OrangeFS version 2.9.8 with GCC version 8.3.0 and LMDB 0.9.22 from the Buster repository. As explained above, OrangeFS has only a single type of daemon, which was running on all nodes with disks. Metadata was stored by the daemon, which was deployed on node C, while the other nodes stored the data. As OrangeFS offers no data redundancy for data that is not read-only, ZFS version 2.0.3 was used to mirror disks locally.

We used MooseFS version 3.0.115, which is available in the Buster backports repository. The chunk servers were deployed on nodes A1–A4, the master server on node C and the monitoring server on node B. As file deletes are always asynchronous in MooseFS, deleted files will be removed in the background after a certain time, called trash time. We made sure to avoid overlapping reads and writes with background file deletions by keeping the default trash time of 24 hours and timed our benchmarks accordingly.

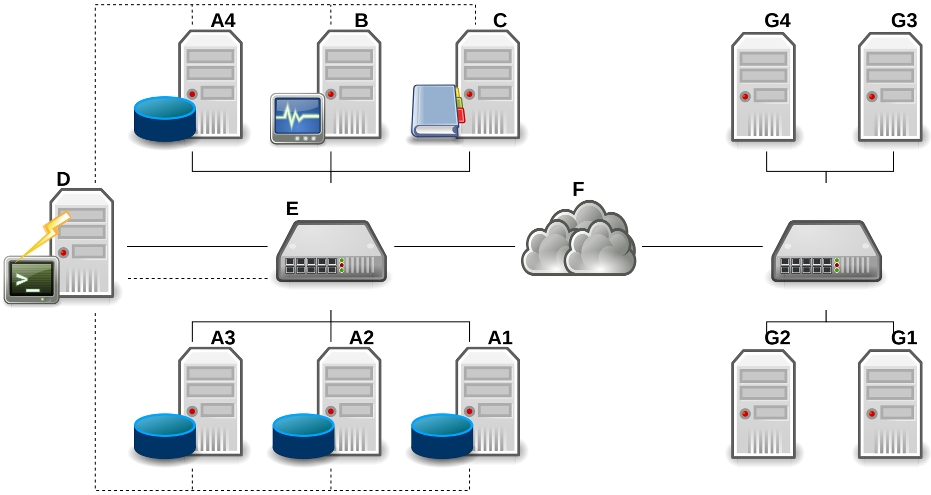


Figure 1. This graphic shows the network topology of the ARM-based cluster. The storage nodes (A1–A4), the management node (B), the metadata node (C), and the BananaPi with the power meter (D) are connected to the Netgear switch (E). The dotted lines indicate which devices are included in power measurements. The storage cluster is connected to the clients (G1–G4) via the Max Planck Institute Magdeburg network infrastructure (F).

We built GlusterFS version 9.2 with GCC version 8.3.0. All nodes with disks were part of a trusted storage pool. We used one disk as brick per node, which was formatted with XFS according to the recommendations from the GlusterFS documentation [15]. Additionally, we exchanged the SSDs on node C for HDDs of the same type as on nodes A1–A4, as GlusterFS cannot benefit from multiple storage tiers within a single volume. The dispersed volume was created using all nodes in the TSP and a redundancy count of one, resulting in a stripe size of 2 KiB. These changes for GlusterFS did not change the theoretical peak performance, which is then bound by the network throughput of the clients.

D. Theoretical Peak Performance

As can be seen in Table I, the theoretical peak performance (TPP) of the ARM cluster is limited by the network throughput of each node, which is not as high as the aggregated throughput of all storage devices of the node. The same applies to reference cluster 1.

Because no measurements could be made in the productive reference cluster (reference cluster 2), the maximum throughput of the components is taken from the respective datasheets. Adding together the TPP of the two node types, its TPP is **44.04 GiB/s**.

This analysis neglects metadata operations, which are, reasonably, assumed not to limit the data throughput of the cluster, for a few files in use. Furthermore, the table only presents the performance for writes. However, as the network already limits peak performance for the ARM cluster, and aggregated throughput of the SSDs in Supermicro nodes of the reference cluster is close to the network speed, the same applies, approximately, to reads.

E. Results

The results of the performance and energy efficiency metrics are shown in Figures 2 to 4. Each measurement was repeated five times and the error bars depict the standard deviation of those samples. The samples for the throughput per Watt metric are computed by dividing each performance measurement by the mean power consumption of its measurement iteration. As explained above, the EDP is normalized by the lowest value per comparison.

Figure 5 shows box plots of the distributions of measured power samples during the last measurement iteration with 4 clients on the ARM cluster, to gain further insight into the power consumption of the different parallel file systems.

The capacity metric shown in Figure 6 was computed using the idle power consumption of the clusters and the raw storage capacity. Nevertheless, the usable storage capacity depends on the respective software setup. The ARM cluster achieved **0.178 TiB/W** and the productive reference cluster (reference 2) **0.066 TiB/W**. Reference cluster 1 is excluded from this metric, because it is not designed as a storage cluster and therefore not equipped with many high capacity storage devices.

VI. Discussion

In this section, we discuss general aspects of our experiments followed by a discussion on the results.

All results need to be seen in relation to the respective systems' cost, as the ARM cluster nodes and disks cost only about € 1,350, while the reference cluster nodes and disks cost around € 40,000. In addition, the reference cluster 2 only uses NVMe SSDs, while the ARM-based cluster uses HDDs for data object storage. Due to the low sampling rate of the power measurements for the reference cluster 2, some spikes in the energy consumption are possibly missed, resulting in an underestimation. In contrast, power measurements on the ARM-based cluster can be expected to overestimate the actual power consumption of the nodes and disks, as only the average idle power consumption of the switch is subtracted.

During previous experiments on a BananaPi M1 single-board computer cluster, the deployment of traditional parallel

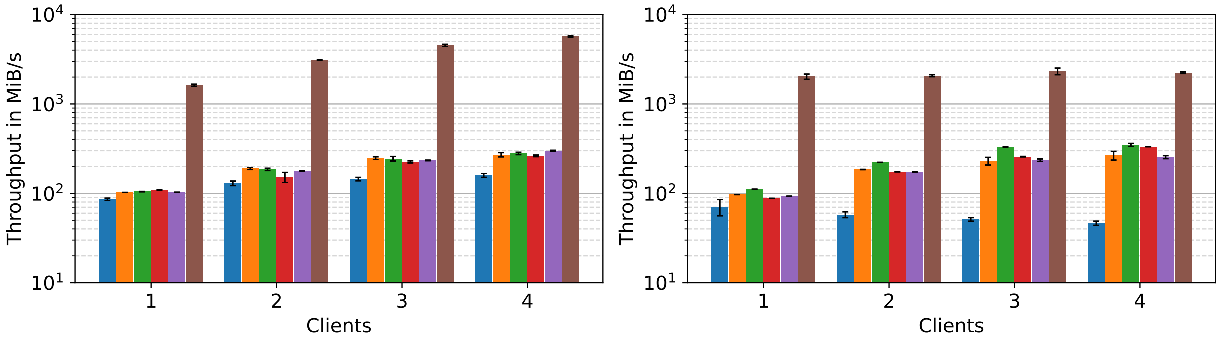


Figure 2. Throughput for reading (left) and writing (right)

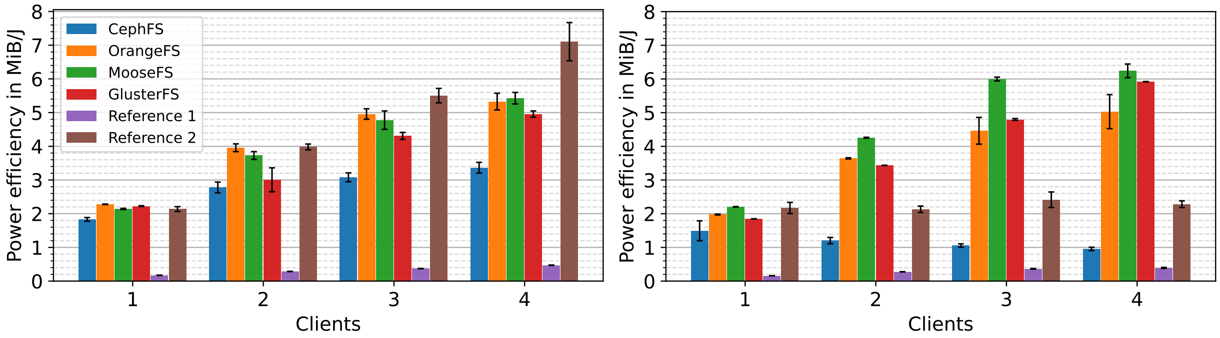


Figure 3. Power efficiency in throughput per Watt for reading (left) and writing (right)

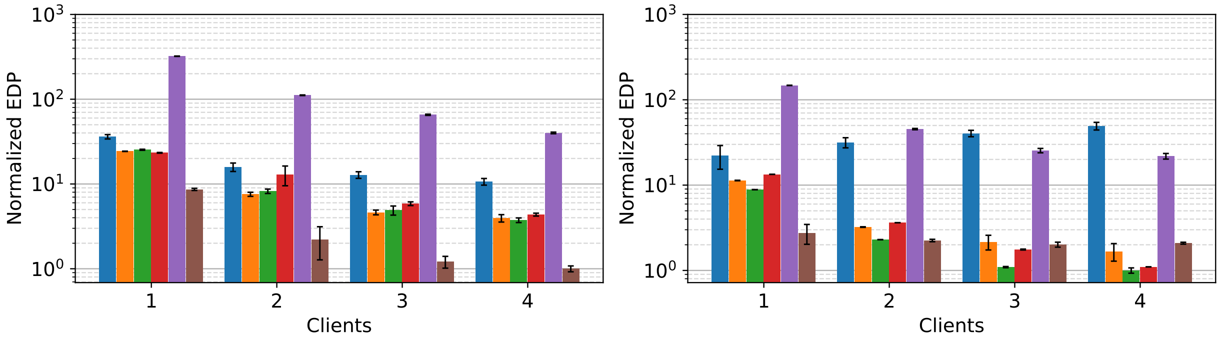


Figure 4. Normalized energy-delay product for reading (left) and writing (right)

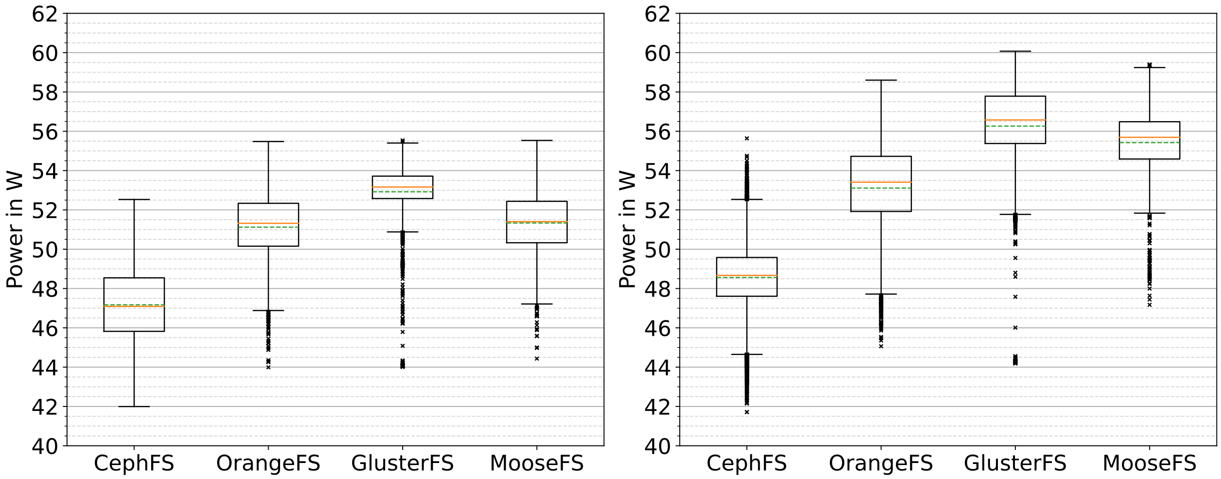


Figure 5. Distribution of power samples for the last measurement iteration with 4 clients for reading (left) and writing (right). In addition to the quartiles, the mean is depicted as green dashed line.

TABLE I. Throughput of components relevant for Theoretical Peak Performance (TPP) Throughput

Cluster	Network	Throughput Storage Devices	Storage Devices per Node	# Nodes	TPP
ARM	111.34 MiB/s	140.09 MiB/s	2	4	445.36 MiB/s
Reference 1	112.22 MiB/s	115.35 MiB/s	1	4	448.88 MiB/s
Reference 2 - Supermicro	11.64 GiB/s	2.70 GiB/s	4	3	32.4 GiB/s
Reference 2 - Gigabyte	11.64 GiB/s	2.70 GiB/s / 3.73 GiB/s	1+8	1	11.64 GiB/s

TABLE II. Maximum throughput of all measurement iterations in MiB/s and percent of TPP.

System	Write / % TPP	Read / % TPP
ARM - CephFS	95.22 / 21.38	172.12 / 38.65
ARM - OrangeFS	289.23 / 64.94	296.82 / 66.65
ARM - MooseFS	365.57 / 82.08	291.41 / 65.43
ARM - GlusterFS	333.26 / 74.83	268.60 / 50.31
Reference 1	266.48 / 59.37	305.52 / 68.06
Reference 2	2322.47 / 5.15	5705.00 / 12.65

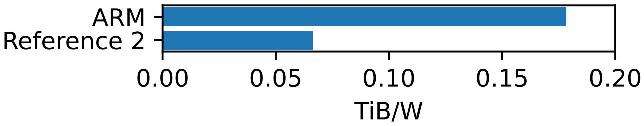


Figure 6. Storage capacity per Watt

file systems proved difficult on the unusual hardware. Tested file systems were CephFS, OrangeFS and BeeGFS. Both CephFS and BeeGFS needed small patches to run on the setup. OrangeFS could not run the client on ARM 32-bit using the upstream kernel module. Additionally, we observed low read throughput if no direct I/O was used. For four clients reading a 2 GiB file each, only 12.41 MiB/s could be achieved. Consequently, measurements on OrangeFS are done with direct I/O.

Our prototype cannot compete with the throughput of the productive reference cluster 2. However, using reference cluster 1 the performance of the ARM-based cluster seems comparable. We used these two references for different purposes. While reference cluster 1 has nearly the same TPP, and is therefore good to compare the performance of the different file systems on different architectures, it is made of legacy hardware and not built as a storage cluster. Because of this, it is not beneficial as a reference for the energy efficiency. Consequently, we used the second reference cluster, which is made of modern hardware, to validate the ARM-based cluster in terms of energy efficiency. For real world HPC applications, more storage nodes would need to be added to the ARM-based cluster to achieve higher throughput. This cluster was built as a proof-of-concept for throughput efficiency and to gain insight in ARM single-board computer storage clusters.

The different read and write sizes on both setups were chosen to achieve reasonable run-times of the benchmarks on both settings. Neither throughput nor throughput efficiency are influenced by the different amounts of transferred data if run-times are long enough.

A. Performance

All clusters show good throughput scaling when adding more clients. Exceptions occur for writes. On the second reference cluster, one client achieves close to the observed maximum performance, and no further improvement can be seen when adding more clients. On the CephFS setup, on the ARM-based cluster, the situation is even worse, as performance drops for more clients. Both Ceph-based systems only reached a fraction of the theoretical peak performance, as can be seen in Table II. For the ARM cluster, this is most likely related to data replication over the public network. This hypothesis is supported by the fact that Ceph OSDs reported slow operation warnings due to waiting times for sub-operations. As pointed out by Just [53], the Ceph OSD service utilizes many threads, which could lead to performance issues for a few cores, as context switches introduce additional overhead. Ceph's behaviour is strongly influenced by the number of placement groups per OSD [3]. While a higher ratio of placement groups to OSDs ensures a balanced data distribution, management of each placement group consumes memory and CPU time. To minimize overhead, we set both pools to 64 placement groups. The number of placement groups per OSD also influences recovery behavior for larger clusters as more placement groups need to be replicated in case of a server crash. Further experiments are needed to evaluate different placement group counts and placement group to OSD ratios for productive usage of Ceph on large ARM clusters.

Nevertheless, replication cannot explain the performance drop for the second reference cluster, which needs further investigation. One impacting factor for reads was that only one process per client was used, resulting in only one network stream, insufficient to saturate the network. This decision was made for comparability with the ARM cluster.

Both Ceph-based systems might be impacted by CephFS' lazy deletes [3], which are done asynchronously by an MDS and probably overlapped with reads and writes, resulting in lower throughput.

OrangeFS performs better than CephFS on ARM in nearly all measurements. In contrast to CephFS, the OrangeFS daemon is lightweight and does not use many threads. Therefore, context switches introduce less overhead on low core counts. Because no replication is done between nodes, less data needs to be transferred via the network, and the management of replicas does not consume resources. The downside is that faults of nodes can lead to data loss. Even though performance is higher compared to CephFS, only about 60 % of the TPP (see Table II) can be achieved.

MooseFS behaved similar to OrangeFS overall. However, it achieved the maximum of write throughput of all measurements at 82.08% of TPP, see Table II. Nevertheless, a disadvantage from the perspective of a single user might be the asynchronous deletion of files in the background, because this overlapping workload reduces the performance of parallel I/O. Different parameters on how background operations are performed are available in MooseFS. Further tuning of the system is needed to show if this problem can be mitigated.

Although GlusterFS is not designed for these parallel coordinated accesses its overall performance was comparable to OrangeFS on the ARM-based cluster. Its performance does not seem to be impaired by the small stripe size of only 2 KiB. However, it also had an advantage compared to the other file systems because one additional server was available for data storage resulting in lower I/O stress per node. On the other hand, due to redundant data blocks added by erasure coding, each node received the same amount of data to write as with OrangeFS and MooseFS.

While most of its volume types are not suitable for HPC workloads, dispersed volumes enable parallel access to multiple servers within one file and ensure data safety using erasure coding. In contrast to replication, erasure coding will not duplicate all of the data blocks, but add redundant blocks. Thus, it is more space efficient and puts less stress on the network than replication.

The disadvantage of this volume type in GlusterFS is that the stripe size depends on the number of bricks in the volume and the desired redundancy count. Scaling up such systems, without changing the stripe size, can be accomplished by combining volume types. Multiple dispersed volumes can be part of a single distributed volume, which would distribute whole files to the different dispersed volumes.

All tested file systems on the ARM cluster can certainly be tuned for higher throughput. Many settings of different storage layers influence their behavior. On top of that, the interactions between the layers are non-trivial. For example, let us look at OrangeFS: Tuning the stripe size and the record size of ZFS can be a first optimization. Compared to the defaults of other parallel file systems, OrangeFS has a relatively low default stripe size of 64 KiB. Further benchmarks should be done to evaluate bigger stripes, which could result in larger disk accesses depending on server-side cache size and cache times. As shown by traces of MPI-IO calls and OrangeFS' internal Trove layer, which does the actual disk I/O, single client-side write calls can result in multiple server-side Trove write calls [54]. Those should align to ZFS record sizes, if possible, to minimize read-modify-write cycles. Additionally other local file systems (e.g., XFS, BTRFS) and layers for local disk mirroring (e.g., LVM, MDADM) could be evaluated.

B. Energy Efficiency

While the ARM-based cluster was hardly comparable with the second reference cluster in terms of performance, its energy efficiency and throughput per Watt was similar to or even exceeded the second reference for all file systems except CephFS. In contrast, the first reference cluster shows devastating energy efficiency. The reason for this is the high energy consumption while only operating on a 1 Gbit/s network and therefore showing similar performance to the ARM-based cluster.

Apart from the first reference cluster, the energy efficiency plots resemble the performance plots in the relations between the file systems. This similarity suggests that energy efficiency on the ARM-based cluster was mostly determined by performance and resulting run-times of the operations.

To determine whether there are more differences between the systems than performance and resulting run-times, we took a look at the measured power consumption during the last measurement iteration with 4 clients, which can be seen in Figure 5. File systems that showed a higher throughput and energy efficiency, also consumed more power during the benchmarks. This pattern indicates that a higher hardware utilization leads to better performance and ultimately higher energy efficiency. CephsFS, for example, was possibly not able to fully utilize the disks on the nodes, due to the OSDs' overhead, which led to lower power consumption of disks, but also to lower throughput. GlusterFS, on the other hand, had the highest power consumption, which, in combination with its high throughput, suggests a high hardware utilization. Even so, GlusterFS' power consumption is slightly higher, here, because two SSDs were swapped for HDDs.

Overall, the ARM cluster's low idle power and maximum power consumption allow for usage of the cluster in places or situations where power restrictions apply, enabling the usage as a mobile storage solution.

In terms of capacity per Watt, the ARM cluster is superior to the second reference cluster, achieving 2.68 times more TB per Watt. However, this result could easily be changed by using higher capacity disks on both clusters. For a more sophisticated comparison between the system architectures in this regard, the power consumption of the server nodes should be measured separated from the disks as done by Gudu and Hardt [26] resulting in a storage controller energy efficiency metric. Nevertheless, this metric is useful for optimizing existing storage solutions.

C. Energy-Delay Product

Compared to the other metrics, the EDP, as shown in Figure 4 is a fused metric that measures performance and energy efficiency at once. The use of this metric for tuning storage systems enforces that balanced configurations are found. Neither performance nor energy-saving efforts are neglected in favor of the other. One example is given by the first reference cluster and CephFS on the ARM-based cluster, see Figure 4. While CephFS had low performance but also a low power consumption its EDP is lower at first. Nevertheless, with more clients the first reference cluster starts to gain advantage because performance starts to out weigh energy consumption.

Even so, for practical applications the weight of the EDP has to be chosen carefully. To evaluate HPC applications one would likely choose higher weights to put more focus on performance. Another problem is imposed by great fluctuations of the measured EDP for repeated measurements, which are even amplified when a larger weight is chosen. This could be mitigated by using shorter benchmarks and more repetitions.

VII. Conclusion and Future Work

We evaluated different file systems for HPC workloads on two reference clusters, based on traditional x86 servers, and an ARM-based low-power cluster. We compared the results in terms of throughput and efficiency. The ARM cluster is able to provide more than twice as much TB per Watt compared to the reference cluster and can achieve similar throughput efficiency. OrangeFS, MooseFS and GlusterFS have been shown to perform better than CephFS on the ARM cluster. Due to the low idle power consumption and low power peaks, ARM-based storage solutions are helpful in situations where power restrictions apply, for example, when used as a mobile storage cluster. In summary, we have shown that the energy efficiency of storage solutions depends significantly on both the used architecture and the file system. Lightweight solutions can reduce energy consumption and thus cost, which is becoming increasingly important due to the exponentially growing volumes of data.

As a next step, we will evaluate the ARM-based cluster using other workloads that are of interest. Examples of such workloads are metadata-focused workloads and mixed workloads that would be produced by multiple users accessing the storage cluster in an uncoordinated manner. Such workloads will show whether ARM-based storage clusters with many small nodes can generally replace traditional storage clusters, or whether they are more suitable for smaller or special purpose systems. For this reason, throughput scaling of the ARM cluster while adding more storage nodes also needs to be measured.

Acknowledgment

This work is partly funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 417705296. More information about the CoSEMoS (Coupled Storage System for Efficient Management of Self-Describing Data Formats) project can be found at https://cosemos.de.

References

[1] T. L. Erxleben, K. Duwe, J. Saak, M. Köhler, and M. Kuhn, “Energy efficiency of parallel file systems on an ARM cluster,” in *ENERGY 2022, The Twelfth International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies*, vol. 12. IARIA, 2022, pp. 42–48.

[2] J. G. Koomey, “Worldwide electricity used in data centers,” *Environmental Research Letters*, vol. 3, no. 3, 2008, DOI: 10.1088/1748-9326/3/3/034008.

[3] Ceph authors and contributors, “Ceph Documentation,” https://docs.ceph.com/en/latest, 2021, [retrieved: 04, 2022].

[4] S. A. Weil, S. A. Brandt, E. L. Miller, D. D. E. Long, and C. Maltzahn, “Ceph: A Scalable, High-Performance Distributed File System,” in *7th Symposium on Operating Systems Design and Implementation (OSDI ’06), November 6-8, Seattle, WA, USA*, B. N. Bershad and J. C. Mogul, Eds. USENIX Association, 2006, pp. 307–320.

[5] C. Wang, K. Mohror, and M. Snir, “File system semantics requirements of hpc applications,” in *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing*, ser. HPDC ’21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 19–30, DOI: 10.1145/3431379.3460637.

[6] M. M. D. Bonnie *et al.*, “OrangeFS: Advancing PVFS,” in *USENIX Conference on File and Storage Technologies (FAST)*, 2011.

[7] OrangeFS Development Team, “OrangeFS Documentation,” http://docs.orangefs.com/, [retrieved: 05, 2022].

[8] K. Duwe and M. Kuhn, “Using Ceph’s BlueStore as Object Storage in HPC Storage Framework,” in *CHEOPS@EuroSys’21*. ACM, 2021, pp. 3:1–3:6, DOI: 10.1145/3439839.3458734.

[9] J. Edge, “The OrangeFS distributed filesystem,” https://lwn.net/Articles/643165/, 2015, [retrieved: 04, 2022].

[10] R. Thakur, W. Gropp, and E. Lusk, “A Case for Using MPI’s Derived Datatypes to Improve I/O Performance,” in *Proceedings of SC98: High Performance Networking and Computing*. ACM Press, November 1998, DOI: 10.1109/SC.1998.10006.

[11] M. Vilayannur, R. Ross, P. Carns, R. Thakur, A. Sivasubramaniam, and M. Kandemir, “On the performance of the POSIX I/O interface to PVFS,” in *12th Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2004. Proceedings.*, 2004, pp. 332–339, DOI: 10.1109/EMPDP.2004.1271463.

[12] A. Kruszona-Zawadzka, “MooseFS 3.0 User’s Manual,” https://moosefs.com/Content/Downloads/moosefs-3-0-users-manual.pdf, 2017, [retrieved: 04, 2022].

[13] Tappest sp. z o.o., “MooseFS Website,” https://moosefs.com, 2022, [retrieved: 05, 2022].

[14] Z. Baojun, P. Ruifang, and Y. Fujun, “Analyzing and improving load balancing algorithm of MooseFS,” *International Journal of Grid and Distributed Computing*, vol. 7, no. 4, pp. 169–176, Aug. 2014, DOI: 10.14257/ijgdc.2014.7.4.16.

[15] GlusterFS Development Team, “GlusterFS Documentation,” http://docs.gluster.org/, [retrieved: 05, 2022].

[16] Z. Ou, B. Pang, Y. Deng, J. K. Nurminen, A. Ylä-Jääski, and P. Hui, “Energy- and Cost-Efficiency Analysis of ARM-Based Clusters,” in *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*. IEEE Computer Society, 2012, pp. 115–123, DOI: 10.1109/CCGrid.2012.84.

[17] E. L. Padoin, D. A. G. de Oliveira, P. Velho, and P. O. A. Navaux, “Evaluating Performance and Energy on ARM-based Clusters for High Performance Computing,” in *41st International Conference on Parallel Processing Workshops, ICPPW 2012, Pittsburgh, PA, USA, September 10-13, 2012*. IEEE Computer Society, 2012, pp. 165–172, DOI: 10.1109/ICPPW.2012.21.

[18] N. Rajovic, A. Rico, N. Puzovic, C. Adeniyi-Jones, and A. Ramírez, “Tibidabo: Making the case for an

ARM-based HPC system," *Future Generation Computer Systems*, vol. 36, pp. 322–334, 2014, DOI: 10.1016/j.future.2013.07.013.

[19] M. Sato *et al.*, "Co-Design for A64FX Manycore Processor and "Fugaku"," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020, pp. 1–15, DOI: 10.1109/SC41405.2020.00051.

[20] D. Göddeke *et al.*, "Energy efficiency vs. performance of the numerical solution of PDEs: An application study on a low-power ARM-based cluster," *Journal of Computational Physics*, vol. 237, pp. 132–150, 2013, DOI: 10.1016/j.jcp.2012.11.031.

[21] S. Brienza, S. E. Cebeci, S. S. Masoumzadeh, H. Hlavacs, Ö. Özkasap, and G. Anastasi, "A Survey on Energy Efficiency in P2P Systems: File Distribution, Content Streaming, and Epidemics," *ACM Computing Surveys*, vol. 48, no. 3, pp. 36:1–36:37, 2016, DOI: 10.1145/2835374.

[22] G. Lefebvre and M. J. Feeley, "Energy efficient peer-to-peer storage," Technical Report TR-2003-17. Department of Computer Science, University of British Columbia, Tech. Rep., 2000.

[23] M. Geveler, B. Reuter, V. Aizinger, D. Göddeke, and S. Turek, "Energy efficiency of the simulation of three-dimensional coastal ocean circulation on modern commodity and mobile processors," *Computer Science - Research and Development*, vol. 31, no. 4, pp. 225–234, 2016, DOI: 10.1007/s00450-016-0324-5.

[24] F. Mantovani *et al.*, "Performance and energy consumption of hpc workloads on a cluster based on arm thunderx2 cpu," *Future Generation Computer Systems*, vol. 112, pp. 800–818, 2020, DOI: 10.1016/j.future.2020.06.033.

[25] M. Ponce *et al.*, "Deploying a top-100 supercomputer for large parallel workloads: The niagara supercomputer," in *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning)*, ser. PEARC '19. New York, NY, USA: Association for Computing Machinery, 2019, DOI: 10.1145/3332186.3332195.

[26] D. Gudu and M. Hardt, "ARM Cluster for Performant and Energy-Efficient Storage," in *Computational Sustainability*, ser. Studies in Computational Intelligence, J. Lässig, K. Kersting, and K. Morik, Eds. Springer, 2016, vol. 645, pp. 265–276, DOI: 10.1007/978-3-319-31858-5_12.

[27] A. Kougkas, A. Fleck, and X.-H. Sun, "Towards Energy Efficient Data Management in HPC: The Open Ethernet Drive Approach," in *2016 1st Joint International Workshop on Parallel Data Storage and data Intensive Scalable Computing Systems (PDSW-DISCS)*, 2016, pp. 43–48, DOI: 10.1109/PDSW-DISCS.2016.012.

[28] L. Zhang, Y. Deng, W. Zhu, J. Zhou, and F. Wang, "Skewly replicating hot data to construct a power-efficient storage cluster," *Journal of Network and Computer Applications*, vol. 50, pp. 168–179, 2015, DOI: 10.1016/j.jnca.2014.06.005.

[29] X. Ruan *et al.*, "ECOS: An energy-efficient cluster storage system," in *2009 IEEE 28th International Performance Computing and Communications Conference*, 2009, pp. 79–86, DOI: 10.1109/PCCC.2009.5403814.

[30] C. Karakoyunlu and J. A. Chandy, "Techniques for an energy aware parallel file system," in *2012 International Green Computing Conference, IGCC 2012, San Jose, CA, USA, June 4-8, 2012*. IEEE Computer Society, 2012, pp. 1–5, DOI: 10.1109/IGCC.2012.6322247.

[31] P. Sehgal, V. Tarasov, and E. Zadok, "Evaluating Performance and Energy in File System Server Workloads," in *8th USENIX Conference on File and Storage Technologies, San Jose, CA, USA, February 23-26, 2010*, R. C. Burns and K. Keeton, Eds. USENIX, 2010, pp. 253–266.

[32] H. Shan and J. Shalf, "Using IOR to Analyze the I/O Performance for HPC Platforms," in *In: Cray User Group Conference (CUG'07)*, 2007.

[33] S. Rivoire, M. A. Shah, P. Ranganathan, C. Kozyrakis, and J. Meza, "Models and metrics to enable energy-efficiency optimizations," *Computer*, vol. 40, no. 12, pp. 39–48, 2007, DOI: 10.1109/MC.2007.436.

[34] M. Horowitz, T. Indermaur, and R. Gonzalez, "Low-power digital design," in *Proceedings of 1994 IEEE Symposium on Low Power Electronics*, 1994, pp. 8–11, DOI: 10.1109/LPE.1994.573184.

[35] D. Chen *et al.*, "Usage centric green performance indicators," *SIGMETRICS Perform. Evaluation Rev.*, vol. 39, no. 3, pp. 92–96, 2011, DOI: 10.1145/2160803.2160868.

[36] J. H. Laros III *et al.*, "Energy Delay Product," in *Energy-Efficient High Performance Computing: Measurement and Tuning*. London: Springer London, 2013, pp. 51–55, DOI: 10.1007/978-1-4471-4492-2_8.

[37] S. Georgiou, M. Kechagia, P. Louridas, and D. Spinellis, "What Are Your Programming Language's Energy-Delay Implications?" in *Proceedings of the 15th International Conference on Mining Software Repositories*, ser. MSR '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 303–313, DOI: 10.1145/3196398.3196414.

[38] Western Digital Corporation, "WD2502ABYS Datasheet," https://products.wdc.com/library/SpecSheet/ENG/2879-701281.pdf?_ga=2.204934934.742886585.1651008103-1759292557.1651008103, 2008, [retrieved: 04, 2022].

[39] Seagate Technology LLC, "Seagte ST3250318AS Product Manual," https://www.seagate.com/staticfiles/support/disc/manuals/desktop/Barracuda%207200.12/100529369c.pdf, 2009, [retrieved: 04, 2022].

[40] Super Micro Computer, Inc., "Supermicro AS 2124BT-HNTR Datasheet," https://www.supermicro.com/en/Aplus/system/2U/2124/AS-2124BT-HNTR.cfm, 2020, [retrieved: 04, 2022].

[41] Intel Corporation, "Intel P4510 Datasheet," https:

//ark.intel.com/content/www/us/en/ark/products/122579/intel-ssd-dc-p4510-series-4-0tb-2-5in-pcie-3-1-x4-3d2-tlc.html, 2018, [retrieved: 04, 2022].

[42] GIGA-BYTE Technology Co., “Gigabyte R282-Z94 Datasheet,” https://www.gigabyte.com/Enterprise/Rack-Server/R282-Z94-rev-100#Specifications, 2021, [retrieved: 04, 2022].

[43] Samsung, “Samsung MZQL23T8HCJS-00A07 Datasheet,” https://semiconductor.samsung.com/ssd/datacenter-ssd/pm9a3/mzql23t8hcjs-00a07/, 2021, [retrieved: 04, 2022].

[44] HARDKERNEL CO., LTD., “Odroid HC4 Datasheet,” https://wiki.odroid.com/odroid-hc4/hardware/hardware, 2021, [retrieved: 04, 2022].

[45] Raspberry Pi Ltd., “Raspberry Pi 4 Model B specifications,” https://www.raspberrypi.com/products/raspberry-pi-4-model-b/, [retrieved: 10, 2022].

[46] Western Digital Corporation, “WD Black WD10SPSX Datasheet,” https://documents.westerndigital.com/content/dam/doc-library/en_us/assets/public/western-digital/product/internal-drives/wd-black-hdd/product-brief-western-digital-wd-black-mobile-hdd.pdf, 2020, [retrieved: 04, 2022].

[47] Samsung, “Samsung V-NAND SSD 860 PRO Datasheet,” https://www.samsung.com/semiconductor/global.semi.static/Samsung_SSD_860_PRO_Data_Sheet_Rev1_1.pdf, 2018, [retrieved: 04, 2022].

[48] NETGEAR, Inc., “Netgear GS110EMX Datasheet,” https://www.netgear.com/images/datasheet/switches/webmanagedswitches/GS110EMX_GS110MX.pdf, 2021, [retrieved: 04, 2022].

[49] Armbian, “Armbian Odroid HC4,” https://www.armbian.com/odroid-hc4/, 2022, [retrieved: 04, 2022].

[50] MEAN WELL, “MW HRP 450-15 Datasheet,” https://www.meanwell.com/webapp/product/search.aspx?prod=HRP-450, 2021, [retrieved: 04, 2022].

[51] ZES ZIMMER Electronic Systems GmbH, “ZES Zimmer LMG 450 Brochure,” https://www.zes.com/en/content/download/286/2473/file/lmg450_prospekt_1002_e.pdf, 2010, [retrieved: 04, 2022].

[52] Dell Inc., “Dell Precision 3650 Tower Hardware Specification,” https://www.delltechnologies.com/asset/en-us/products/workstations/technical-support/precision-3650-spec-sheet.pdf, 2021, [retrieved: 04, 2022].

[53] S. Just, “Crimson: A new ceph OSD for the age of persistent memory and fast NVMe storage,” Presentation at Linux Storage and Filesystems Conference (Vault ’20), Santa Clara, CA, Feb. 2020, https://www.usenix.org/conference/vault20/presentation/just [retrieved: 04, 2022].

[54] T. Ludwig, S. Krempel, J. Kunkel, F. Panse, and D. Withanage, “Tracing the MPI-IO Calls’ Disk Accesses,” in *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, B. Mohr, J. L. Träff, J. Worringen, and J. Dongarra, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 322–330, DOI: 10.1007/11846802_45.

Using Locally Weighted Regression to Estimate the Functional Size of Software: an Empirical Study

Luigi Lavazza Angela Locoro
Dipartimento di Scienze Teoriche e Applicate
Università degli Studi dell'Insubria
Varese, Italy
email:luigi.lavazza, angela.locoro@uninsubria.it

Geng Liu
Hangzhou Dianzi University
Hangzhou, China
email:liugeng@hdu.edu.cn

Roberto Meli
DPO
Rome, Italy
email:roberto.meli@dpo.it

***Abstract*—In software engineering, measuring software functional size via the IFPUG (International Function Point Users Group) Function Point Analysis using the standard manual process can be a long and expensive activity, which is possible only when functional user requirements are known completely and in detail. To solve this problem, several early estimation methods have been proposed and have become *de facto* standard processes. Among these, a prominent one is High-level Function Point Analysis. Recently, the Simple Function Point method has been released by IFPUG; although it is a proper measurement method, it has a great level of convertibility to traditional Function Points and may be used as an estimation method. Both High-level Function Point Analysis and Simple Function Point skip the activities needed to weight data and transaction functions, thus enabling lightweight measurement based on coarse-grained requirements specifications. This makes the process faster and cheaper, but yields approximate measures. The accuracy of the mentioned method has been evaluated, also via large-scale empirical studies, showing that the yielded approximate measures are sufficiently accurate for practical usage. In this paper, locally weighted regression is applied to the problem outlined above. This empirical study shows that estimates obtained via locally weighted regression are more accurate than those obtained via High-level Function Point Analysis, but are not substantially better than those yielded by alternative estimation methods using linear regression. The Simple Function Point method appears to yield measures that are well correlated with those obtained via standard measurement. In conclusion, locally weighted regression appears to be effective and accurate enough for estimating software functional size.**

***Index Terms*—Function Point Analysis, Early Size Estimation, High-level FPA, Simple Function Points, LOcally Estimated Scatterplot Smoothing (LOESS)**

I. Introduction

This paper extends a previous study that examined a single functional measure dataset [1].

In the late seventies, Allan Albrecht introduced Function Points Analysis (FPA) at IBM [2], as a means to measure the functional size of software, with special reference to the "functional content" delivered by software providers. Albrecht aimed at defining a measure that might be correlated to the value of software from the perspective of a user, and could also be useful to assess the cost of developing software applications, based on functional user requirements.

FPA is a Functional Size Measurement Method (FSMM), compliant with the ISO/IEC 14143 standard, for measuring the size of a software application in the early stages of a project, generally before actual development starts. Accordingly, software size measures expressed in Function Points (FP) are often used for cost estimation.

The International Function Points User Group (IFPUG) is an association that keeps FPA up to date, publishes the official FP counting manual [3], and certifies professional FP counters. Unfortunately, in some conditions, performing the standard IFPUG measurement process may be too long and expensive, with respect to management needs, because standard FP measurement can be performed only when relatively complete and detailed requirements specifications are available, while functional measures could be needed much earlier for management purposes.

Many methods were invented and used to provide *estimates* of functional size measures, based on fewer or coarser-grained information than required by standard FPA. These methods are applied very early in software projects, even before deciding what process (e.g., agile or waterfall) will be used. One of these methods is the High-level FPA (HLFPA) method [4], which was developed by NESMA under the name of "NESMA estimated" method [5].

In 2010, a new FSMM called Simple Function Point (SiFP) was developed by Meli [6]. In 2019, IFPUG acquired the method and in 2021 the IFPUG branded Simple Function Point (SFP) method was delivered to the market [7].

HLFPA and SiFP have been evaluated by several studies, which found that the methods are usable in practice to approximate traditional FPA values, since they yield reasonably accurate estimates. However, the question if it is possible to get more accurate estimates from the basic information used by HLFPA remains open.

In this paper, we evaluate—via an empirical study—the usage of LOESS (LOcally Estimated Scatterplot Smoothing)—also known as LOWESS (LOcally WEighted Scatterplot Smoothing)—to build models that can be used for early estimation of functional size.

We also compare the standard IFPUG FPA measures, the estimates obtained via HLFPA and the estimates obtained via alternative methods (linear regression models and LOESS models) with the measures obtained via the Simple Function Point (SFP) method. SFP is a lightweight method that has also been adopted by IFPUG as an alternative to full-fledged FPA. SFP measurement requires even less time and effort than HLFPA, and it usually yields measures that are very well correlated with IFPUG standard measures.

The work presented here extends previous work [1] by using two datasets to evaluate functional size estimation methods. Specifically, the availability of two datasets allows for cross-dataset evaluations. That is, one dataset is used as the training set, and the other one is used as the test set. This is particularly interesting for practitioners that do not own historical data: our results show that by using a "foreign" dataset for training, it is possible to obtain estimates that appear accurate enough for being used in practice.

In general, the findings reported in this paper contribute to increase our knowledge of the techniques that are available for functional size estimation, their applicability conditions, and the accuracy of the results that can be expected.

The remainder of the paper is organized as follows. Section II provides an overview of functional size measurement methods, and other background information. Section III describes the empirical study and its results. In Section IV the results obtained in the empirical study are discussed, from the technical and managerial points of view. In Section V, we discuss the threats to the validity of the study. Section VI reports about related work. Finally, in Section VII, we draw some conclusions and outline future work.

Note that FPA defines both unadjusted FP (UFP) and adjusted FP. The former are a measure of functional requirements. The latter are obtained by correcting unadjusted FP in order to get an indicator that is expected to be better correlated to development effort. Noticeably, the ISO standardized only unadjusted FP, recognizing UFP as a proper measure of functional requirements [8]. Following the ISO, in this paper we deal only with UFP, even when we speak generically of Function Points or FP. As a matter of fact, also HLFPA aims at providing measures that are compatible with UFP, and not with adjusted FP.

II. Background

Function Point Analysis was originally introduced by Albrecht to measure the size of data-processing systems from the point of view of end-users, with the goal of estimating the value of an application and the development effort [2]. The fortunes of this measure led to the creation of the IFPUG (International Function Points User Group), which maintains the method and certifies professional measurers.

The "amount of functionality" released to the user can be evaluated by taking into account 1) the data used by the application to provide the required functions, and 2) the transactions (i.e., operations that involve data crossing the boundaries of the application) through which the functionality is delivered to the user. Both data and transactions are counted on the basis of Functional User Requirements (FURs) specifications, and constitute the IFPUG Function Points measure.

FURs are modeled as a set of base functional components (BFCs), which are the measurable elements of FURs: each of the identified BFCs is measured, and the size of the application is obtained as the sum of the sizes of BFCs. IFPUG BFCs are: data functions (also known as logical files), which are classified into internal logical files (ILF) and external interface files (EIF); and elementary processes (EP)—also known as transaction functions—which are classified into external inputs (EI), external outputs (EO), and external inquiries (EQ), according to the activities carried out within the considered process and the primary intent.

The complexity of a data function (ILF or EIF) depends on the Record Element Types (RETs), which indicate how many types of variations (e.g., sub-classes, in object-oriented terms) exist per logical data file, and Data Element Types (DETs), which indicate how many types of elementary information (e.g., attributes, in object-oriented terms) are contained in the given logical data file.

The complexity of a transaction depends on the number of FTRs—i.e., the number of File Types Referenced while performing the required operation—and the number of DETs—i.e., the number of types of elementary data—that the considered transaction sends and receives across the boundaries of the application. Details concerning the determination of complexity can be found in the official documentation [3].

The core of FPA involves three main activities:

1) Identifying data and transaction functions.
2) Classifying data functions as ILF or EIF and transactions as EI, EO or EQ.
3) Determining the complexity of each data or transaction function.

The first two of these activities can be carried out even if the FURs have not yet been fully detailed. On the contrary, activity 3 requires that all details are available, so that FP measurers can determine the number of RET or FTR and DET involved in every function. Activity 3 is relatively time- and effort-consuming [9].

HLFPA does not require activity 3, thus allowing for size estimation when FURs are not fully detailed: it only requires that the complete sets of data and transaction functions are identified and classified.

The SFP method [7] does not require activities 2 and 3: it only requires that the complete sets of data and transaction functions are identified.

Both the HLFPA and SFP methods let measurers skip the most time- and effort-consuming activity, which also needs that requirements are fully specified; thus both methods are relatively fast and cheap. The SFP method does not even require classification, making size estimation even faster and less subjective (since different measurers can sometimes classify differently the same transaction, based on the subjective perception of the transaction's primary intent).

A. The High-level FPA method

NESMA defined two size estimation methods: the 'NESMA Indicative' and the 'NESMA Estimated' methods. IFPUG adopted these methods as early function point analysis methods, under the names of 'Indicative FPA' and 'High-level FPA,' respectively [4]. The Indicative FPA method proved definitely less accurate [10], [11]. Hence, in this paper, we consider only the High-level FPA method.

The High-level FPA method requires the identification and classification of all data and transaction functions, but does not require the assessment of the complexity of functions: ILF and EIF are assumed to be of low complexity, while EI, EQ and EO are assumed to be of average complexity. Hence, estimated size is computed as follows:

$$EstSize_{UFP} = 7\ \#ILF + 5\ \#EIF + 4\ \#EI + 5\ \#EO + 4\ \#EQ \quad (1)$$

where *#ILF* is the number of data functions of type ILF, *#EI* is the number of transaction functions of type EI, etc.

B. The Simple Function Point Method

The Simple Function Point measurement method [6] [7] has been specifically designed to be agile, fast, lightweight, easy to use, and with minimal impact on software development processes. It is easy to learn and provides reliable, repeatable, and objective results. Like IFPUG FPA, it is independent of the technologies used and technical design principles.

SFP requires only the identification of Elementary Processes (EP) and Logical Files (LF), based on the following assumptions: 1) a user gives value to a BFC as a whole independently of internal organization and details, and 2) a cost model based on SFP shows a precision that is comparable to that of a cost model based on a detailed FPA measure. The latter assumption has been verified by different studies [12] [13].

SFP assigns a numeric value directly to these BFCs:

$$SFP = 7\ \#LF + 4.6\ \#EP \quad (2)$$

thus significantly speeding up the functional sizing process, at the expense of ignoring the domain data model, and the primary intent of each Elementary Process.

The weights for each BFC were originally given to achieve the best possible approximation of FPA but as long as the method has become a measurement method, those weights became constants, which are not subject to update or change for approximation reasons and that are crystallized for stability, repeatability and comparability reasons. We can approximate the FPA by setting $EstSize_{UFP} = SFP$.

III. Empirical study

A. The Datasets

In the empirical study, we use two datasets. The first is an ISBSG dataset [14], which was also used previously to evaluate SFP [12]; this is the dataset we used in our original work [1].

The second dataset includes data from software projects developed and used by a Chinese financial enterprise (hence, sometimes we make reference to this dataset as the "Chinese" dataset). These data are subject to non-disclosure agreement, therefore we cannot publish them in a replication package. Also the Chinese dataset was used previously [15], [16] in studies concerning the estimation of functional size measures.

Both datasets contain several small project data. As a matter of fact, estimating the size of small projects is not very interesting. Therefore, we removed from the dataset the projects smaller than 200 UFP. The resulting ISBSG dataset includes data from 110 projects having size in the [207, 4202] range. Some descriptive statistics for this dataset are given in Table I (where all values are rounded to integer).

TABLE I
Descriptive statistics for the ISBSG dataset.

	UFP	HLFPA	SFP	*#EI*	*#EO*	*#EQ*	*#ILF*	*#EIF*	*#LF*	*#EP*
Mean	976	888	971	43	46	46	26	24	50	135
StDev	842	739	785	38	71	51	22	23	39	123
Median	639	607	674	29	17	32	20	18	37	82
Min	207	202	223	0	0	0	0	1	12	14
Max	4202	3755	4257	204	442	366	100	172	234	656

While the ISBSG dataset contains data form projects not greater than 4202 UFP, the Chinese dataset contains data also from much larger projects (up to a few thousands UFP). However, to make the results obtained with the two datasets comparable, we used a subset of the Chinese dataset, so that the size range covered by the two datasets is the same. Some descriptive statistics of the resulting dataset (which accounts for 276 projects) are given in Table II (where all values are rounded to integer).

TABLE II
Descriptive statistics for the Chinese dataset.

	UFP	HLFPA	SFP	*#EI*	*#EO*	*#EQ*	*#ILF*	*#EIF*	*#LF*	*#EP*
Mean	1357	1323	1452	34	14	111	44	87	48	242
Sd	1040	1038	1141	39	23	101	60	101	52	200
Median	1041	984	1074	21	4	80	24	52	29	171
Min	200	142	154	0	0	0	0	0	0	2
Max	4079	4689	5349	220	144	524	428	712	276	997

B. Method used

We built models of functional size using LOESS (locally estimated scatterplot smoothing) [17]. LOESS belongs to the family of computational methods, based on least squares regression, for the estimation of functions fitting subsets of points of a dataset, without the need to yield a global function as a model. The way it works is capturing the local variability of neighbour points of the current point analyzed, in order to build up a function that describes the deterministic part of the variation in the data, point by point. For this reason, it is said to combine k-nearest-neighbor-based models into a meta-model. The regression can be linear and non-linear, i.e., polynomial. The mechanism of neighbours selection depends on a smoothing parameter, α, which determines the inclusion span of point neighbours to be included in the fitting polynomial function. A polynomial function of zero degree turns the LOESS curve method into a mobile average smoothing curve. A weighted variant of LOESS is called LOWESS, which stands for "locally weighted scatterplot smoothing". In this variant, local points are weighted for relevance with respect to the analyzed point, which is proportional to the variance brought by each point, with the nearest point receiving more importance and the furthest ones having less importance during models fitting.

C. Procedure

The analysis was carried out using the R programming language and environment [18]. Specifically, we used the `loess` function from the `Stats` package, which is provided as part of the system libraries.

Through the `span` parameter, the `loess` function makes it possible to control the degree of smoothing. In the empirical study, we tried different values for the `span` parameter, namely 0.5, 0.75 and 0.95.

We aimed at building models using the same five variables (*#EI*, *#EO*, *#EQ*, *#ILF*, *#EIF*) used by HLFPA. However, the `loess` function from the `Stats` package does not allow more than 4 independent variables. To overcome this problem, we observe that in the HLFPA method, *#EI* and *#EQ* get the same weight; therefore, it is conceivable to consider EIs and EQs as a single class of transactions (only as far as size estimation is concerned). Accordingly, for each project we compute $\#EIQ = \#EI + \#EQ$. Then we use four independent variables (*#EO*, *#EIQ*, *#ILF*, *#EIF*) to build size models via LOESS. In addition, we built models that use the same two variables (*#LF* and *#EP*) used by SFP. We also built Ordinary Least Square (OLS) linear regression models.

The evaluation was carried out via 10-time 10-fold cross validation. For all the estimates obtained from 10-time 10-fold cross validation, we compute estimation errors and a few indicators, as follows. The error (alias residual) for the i^{th} estimation is defined as $ee_i = S_i - E_i$, where S_i is the actual size of the element involved in the i^{th} estimation (i.e., the size measured according to the IFPUG standard process) and E_i is the estimated size. The computed indicators are:

- MAR is the Mean of Absolute Residuals, i.e., $MAR = \frac{1}{n}\sum_{i=0}^{n}|ee_i|$, where n is the number of estimates.
- MR is the MAR divided by the mean size $\frac{1}{n}\sum_{i=0}^{n}S_i$. It gives an idea of the relative importance or the estimation errors.
- MdAR is the median of absolute residuals.
- MdR is MdAR divided by the median size. It gives an idea of the relative importance or the estimation errors, while taking into account that the distribution of sizes is skewed.
- MMRE is the mean magnitude of relative errors. $MMRE = \frac{1}{n}\sum_{i=0}^{n}|re_i|$, where $re_i = \frac{ee_i}{S_i}$ is the relative error. MMRE has been widely criticized as a biased metric [19]: we report it for completeness. At any rate, we also report MR, which is not a biased metric, since the mean size is a characteristic of the given dataset: MR is a sort of normalization of the MAR.
- MdMRE is the median magnitude of relative errors.
- Finally, R^2 (the coefficient of determination) is given, since it is a quite reliable indicator of the models' accuracy [20].

To assess the effect size, we use the non-parametric statistic A by Vargha and Delaney [21], as provided by the R package `effsize` [22].

To evaluate if the estimates provided by a method are significantly better than those provided by another method, we tested the statistical significance of the differences among absolute errors yielded by the considered methods [19]. Namely, we compared the absolute residuals via Wilcoxon sign rank test [23] (using the `wilcox.test` function from the R `Stats` package).

D. Evaluation procedure

Our study was carried out in two steps, the first one dealing with within-dataset and the second one with cross-dataset evaluation.

The within-dataset evaluation was carried out using the ISBSG dataset (as reported [1]) and the Chinese dataset. In both cases, we carried out 10-times 10-fold cross validation. In the process, we did not always get usable results. Specifically, via OLS regression we sometimes obtained invalid models (e.g., models with not normally distributed residuals); via LOESS we obtained models that did not support estimation in extreme cases, i.e., for too large or too small independent variables. All these cases were not evaluated. They are a strict minority, hence the reported results represent the most likely outcome of estimation in practice.

Cross-dataset evaluation was straightforward: we built a model (for each of the considered types) using the ISBSG dataset as the training set, and evaluated it using the Chinese dataset as the test set. This operation was then repeated using the Chinese dataset for training and the ISBSG dataset for testing.

E. Results of within-dataset evaluations

This section reports the results obtained for the within-dataset evaluations obtained using first the ISBSG dataset, and then the Chinese dataset.

Results obtained with the ISBSG dataset

The accuracy indicators computed over the estimates that were obtained for the ISBSG dataset are given in Table III. Models LMv are built using OLS regression using v independent variables; models LWMv (where LWM stands for Locally Weighted Model) are built using LOESS, based on v independent variables. For LWMv we give in parentheses the value of the span value.

Table III suggests that OLS linear models provide quite good estimates. Surprisingly, LM4, i.e., the model based on *#EO*, *#EIQ*, *#ILF*, *#EIF* achieves better results than the LM5, i.e., the model based on *#EO*, *#EI*, *#EQ*, *#ILF*, *#EIF*.

We can also observe that estimation accuracy of LWM models varies with the `span`; specifically, accuracy improves with `span`. However, the improvement is modest for LWM2 (MAR decreases from 91.4 to 86.6), while it is quite large for LWM4 (MAR decreases from 93.7 to 55.6). Overall, it seems that when LOESS is used with two variables it is not

TABLE III
WITHIN-DATASET EVALUATION USING THE ISBSG DATASET: ACCURACY INDICATORS.

	MAR	MR	MdAR	MdR	MMRE	MdMRE	R^2
HLFPA	103.8	0.106	58.0	0.091	0.097	0.084	0.966
SFP	87.1	0.089	60.5	0.095	0.105	0.078	0.978
LM5	62.0	0.064	40.6	0.064	0.074	0.057	0.985
LM4	58.2	0.060	39.0	0.061	0.071	0.055	0.987
LM2	91.6	0.096	52.2	0.089	0.096	0.084	0.971
LWM4(0.5)	93.7	0.107	53.5	0.089	0.109	0.089	0.943
LWM2(0.5)	91.4	0.099	56.5	0.089	0.103	0.082	0.940
LWM4(0.75)	66.5	0.076	39.5	0.066	0.082	0.068	0.972
LWM2(0.75)	88.7	0.096	58.2	0.091	0.101	0.075	0.950
LWM4(0.95)	55.6	0.064	37.4	0.062	0.073	0.064	0.984
LWM2(0.95)	86.6	0.094	53.9	0.085	0.096	0.072	0.958

able to substantially improve the estimates provided by LM2; instead, LOESS used with four variables achieves good results, provided that span is sufficiently large. In fact, the minimum MAR is achieved by LWM4 with span=0.95.

It can also be observed that SFP measures provide an approximation that is better than HLFPA's, and not much worse than the best estimators'. Considering that SFP uses fixed weights and does not even require classifying data and transactions, and that the method is not specifically intended to approximate IFPUG measures, this is a quite remarkable result.

The results of the Wilcoxon sign rank test (which are all statistically significant at the usual $\alpha = 0.05$ level) are given in Table IV, where symbol “$>$” (respectively, “$<$” and “$=$”) in the cell at row i and column j indicates that the model in row i has greater (respectively, smaller and equal) absolute residuals than the model in column j.

TABLE IV
WITHIN-DATASET EVALUATION USING THE ISBSG DATASET: COMPARISON OF MODELS' ABSOLUTE RESIDUALS VIA WILCOXON SIGN RANK TEST.

	HLFPA	SFP	LM5	LM4	LM2	LWM4 (0.5)	LWM2 (0.5)	LWM4 (0.75)	LWM2 (0.75)	LWM4 (0.95)	LWM2 (0.95)
HLFPA	–	>	>	>	>	>	>	>	>	>	>
SFP	<	–	>	>	>	=	>	>	>	>	>
LM5	<	<	–	>	<	<	<	<	<	>	<
LM4	<	<	<	–	<	<	<	<	<	<	<
LM2	<	<	>	>	–	<	<	>	=	>	>
LWM4(.5)	<	=	>	>	>	–	=	>	>	>	>
LWM2(.5)	<	<	>	>	>	=	–	>	>	>	>
LWM4(.75)	<	<	>	>	<	<	<	–	<	>	<
LWM2(.75)	<	<	>	>	=	<	<	>	–	>	>
LWM4(.95)	<	<	<	>	<	<	<	<	<	–	<
LWM2(.95)	<	<	>	>	<	<	<	>	<	>	–

To assess the effect size, we use the non-parametric statistic A by Vargha and Delaney [21], as provided by the R package effsize [22]. We obtained the results given in Table V, where each numeric result is accompanied by its interpretation [22]: ‘n’ and ‘s’ indicate negligible and small effect size, respectively.

LWM4(0.95) appears to be the best model according to MAR (Table III). However, According to the Wilcoxon sign rank test, LM4 is the most accurate model. The disagreement between this two indications is explained by Vargha and Delaney's A, which is 0.51 for LM4 vs. LWM4(0.95), showing that the size effect is practically nil, i.e., LM4 is better, but by a practically irrelevant extent.

Finally, we look into the error distributions yielded by the estimation methods that we used in the study.

Figure 1 shows the boxplots of estimation errors for each of the used methods. It can be noticed that LWM2 models provide exceedingly large errors in a few cases.

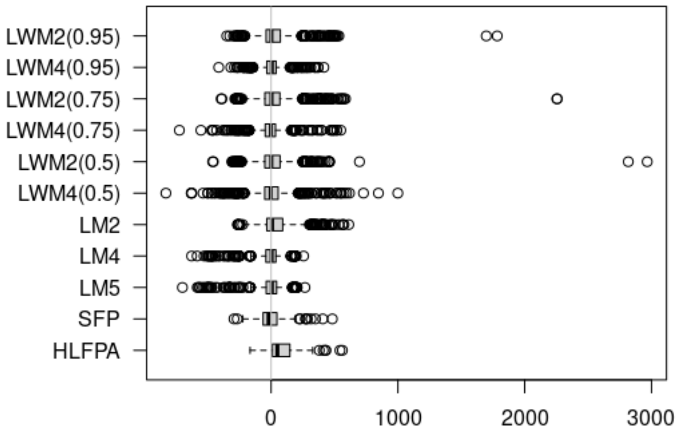


Fig. 1. Within-dataset evaluation using the ISBSG dataset: error boxplots.

Figure 2 provides the same information as Figure 1, but omitting outliers. It can be seen that the various models do not yield dramatically different accuracy levels, when the outliers are excluded. However, it is noteworthy that HLFPA tends to underestimate (as already noted in [16]). The other models provide more balanced errors, with medians very close to zero.

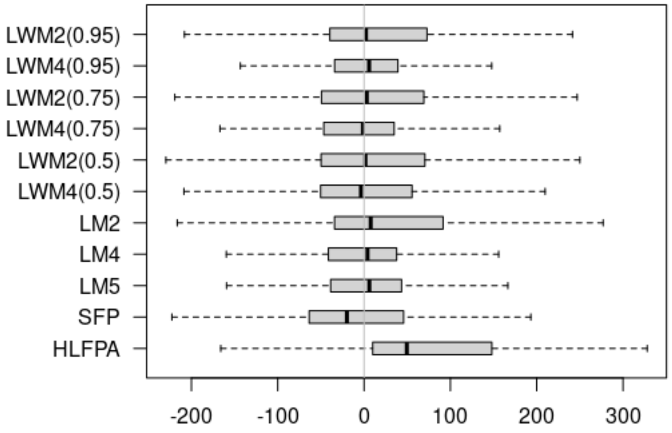


Fig. 2. Within-dataset evaluation using the ISBSG dataset: error boxplots (no outliers).

Figure 3 shows the boxplots of absolute estimation errors for each of the used methods, excluding outliers. The mean absolute error (i.e., the MAR) is shown as an orange diamond. Also according to Figure 3, LM4, LM5 and LWM4(0.95) are the most accurate models.

TABLE V
WITHIN-DATASET EVALUATION USING THE ISBSG DATASET: EFFECT SIZE ACCORDING TO VARGHA AND DELANEY'S A.

	HLFPA	SFP	LM5	LM4	LM2	LWM4 (0.5)	LWM2 (0.5)	LWM4 (0.75)	LWM2 (0.75)	LWM4 (0.95)	LWM2 (0.95)
HLFPA	NA	0.52(n)	0.61(s)	0.62(s)	0.54(n)	0.53(n)	0.53(n)	0.59(s)	0.55(n)	0.61(s)	0.56(n)
SFP	0.48(n)	NA	0.60(s)	0.62(s)	0.52(n)	0.51(n)	0.51(n)	0.58(s)	0.53(n)	0.60(s)	0.54(n)
LM5	0.39(s)	0.40(s)	NA	0.52(n)	0.44(n)	0.42(s)	0.42(s)	0.49(n)	0.43(n)	0.51(n)	0.45(n)
LM4	0.38(s)	0.38(s)	0.48(n)	NA	0.42(s)	0.40(s)	0.40(s)	0.47(n)	0.42(s)	0.49(n)	0.43(n)
LM2	0.46(n)	0.48(n)	0.56(n)	0.58(s)	NA	0.48(n)	0.49(n)	0.55(n)	0.50(n)	0.57(n)	0.51(n)
LWM4(0.5)	0.47(n)	0.49(n)	0.58(s)	0.60(s)	0.52(n)	NA	0.50(n)	0.57(n)	0.52(n)	0.59(s)	0.53(n)
LWM2(0.5)	0.47(n)	0.49(n)	0.58(s)	0.60(s)	0.51(n)	0.50(n)	NA	0.56(n)	0.51(n)	0.58(s)	0.52(n)
LWM4(0.75)	0.41(s)	0.42(s)	0.51(n)	0.53(n)	0.45(n)	0.43(n)	0.44(n)	NA	0.45(n)	0.52(n)	0.47(n)
LWM2(0.75)	0.45(n)	0.47(n)	0.57(n)	0.58(s)	0.50(n)	0.48(n)	0.49(n)	0.55(n)	NA	0.57(n)	0.51(n)
LWM4(0.95)	0.39(s)	0.40(s)	0.49(n)	0.51(n)	0.43(n)	0.41(s)	0.42(s)	0.48(n)	0.43(n)	NA	0.45(n)
LWM2(0.95)	0.44(n)	0.46(n)	0.55(n)	0.57(n)	0.49(n)	0.47(n)	0.48(n)	0.53(n)	0.49(n)	0.55(n)	NA

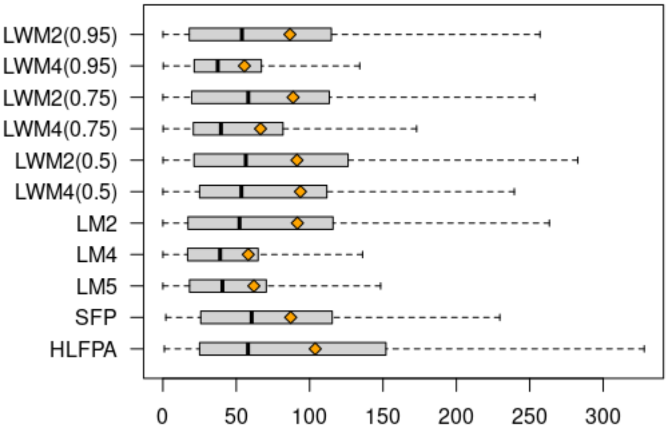


Fig. 3. Within-dataset evaluation using the ISBSG dataset: absolute error boxplots (no outliers).

Results obtained with the Chinese dataset

The accuracy indicators computed over the estimates that were obtained for the Chinese dataset are given in Table VI.

TABLE VI
WITHIN-DATASET EVALUATION USING THE CHINESE DATASET: ACCURACY INDICATORS.

	MAR	MR	MdAR	MdR	MMRE	MdMRE	R^2
HLFPA	119.0	0.088	69.0	0.066	0.095	0.077	0.970
SFP	154.3	0.114	91.9	0.088	0.124	0.108	0.945
LM5	121.0	0.089	78.1	0.076	0.104	0.087	0.972
LM4	128.3	0.095	75.7	0.074	0.105	0.087	0.964
LM2	131.8	0.097	75.8	0.073	0.108	0.088	0.960
LWM4(0.5)	151.9	0.115	82.3	0.081	0.119	0.098	0.942
LWM2(0.5)	116.6	0.087	69.3	0.068	0.104	0.083	0.970
LWM4(0.75)	154.7	0.117	79.9	0.079	0.120	0.104	0.939
LWM2(0.75)	118.7	0.089	74.9	0.073	0.104	0.083	0.970
LWM4(0.95)	123.6	0.094	74.9	0.074	0.106	0.090	0.966
LWM2(0.95)	118.8	0.089	77.8	0.076	0.104	0.082	0.970

Table VI shows that HLFPA provides quite good estimates: better than those achieved for the ISBSG dataset, according to MR. OLS linear models provide estimates that are slightly less accurate than HLFPA's; as expected, the fewer independent variables are used, the less accurate the estimates. Surprisingly, models LM4 (regardless `span`) perform worse than LMW2, which achieve the smallest MAR.

The results of the Wilcoxon sign rank test (which are all statistically significant at the usual $\alpha = 0.05$ level) are given in Table VII.

TABLE VII
WITHIN-DATASET EVALUATION OF THE CHINESE DATASET: COMPARISON OF MODELS' ABSOLUTE RESIDUALS VIA WILCOXON SIGN RANK TEST.

	HLFPA	SFP	LM5	LM4	LM2	LWM4 (0.5)	LWM2 (0.5)	LWM4 (0.75)	LWM2 (0.75)	LWM4 (0.95)	LWM2 (0.95)
HLFPA	–	<	<	<	<	<	=	<	<	<	<
SFP	>	–	>	>	>	=	>	>	>	>	>
LM5	>	<	–	>	>	<	>	<	>	>	>
LM4	>	<	<	–	=	<	>	<	>	>	>
LM2	>	<	<	=	–	<	>	<	=	=	>
LWM4(0.5)	>	=	>	>	>	–	>	=	>	>	>
LWM2(0.5)	=	<	<	<	<	<	–	<	<	<	<
LWM4(0.75)	>	<	>	>	>	=	>	–	>	>	>
LWM2(0.75)	>	<	<	<	=	<	>	<	–	=	>
LWM4(0.95)	>	<	<	<	=	<	>	<	=	–	>
LWM2(0.95)	>	<	<	<	<	<	>	<	<	<	–

According to the Wilcoxon sign rank test, HLFPA provides smaller absolute errors than all other models, except for LWM2(0.5). At any rate, Vargha and Delaney's A, indicates that all model pairs are likely to provide very similar absolute residuals. Finally, we look into the error distributions yielded by the estimation methods that we used in the study. Figure 4 shows the boxplots of estimation errors for each of the used methods. The same boxplots are shown in Figure 5 without outliers, to improve readability. It can be noticed

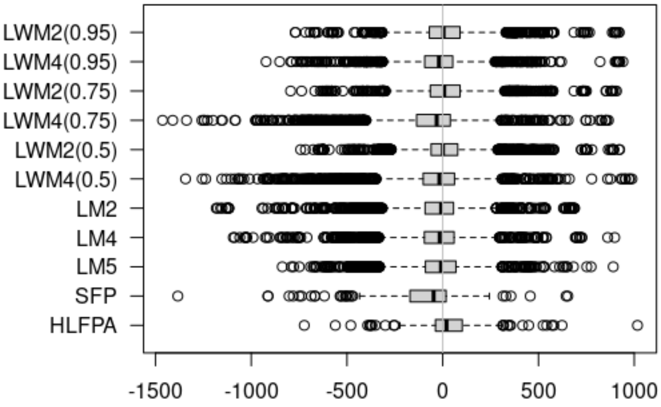


Fig. 4. Within-dataset evaluation using the Chinese dataset: error boxplots.

TABLE VIII
WITHIN-DATASET EVALUATION USING THE CHINESE DATASET: EFFECT SIZE ACCORDING TO VARGHA AND DELANEY'S A.

	HLFPA	SFP	LM5	LM4	LM2	LWM4 (0.5)	LWM2 (0.5)	LWM4 (0.75)	LWM2 (0.75)	LWM4 (0.95)	LWM2 (0.95)
HLFPA	NA	0.45(n)	0.48(n)	0.48(n)	0.49(n)	0.46(n)	0.50(n)	0.46(n)	0.49(n)	0.49(n)	0.49(n)
SFP	0.55(n)	NA	0.53(n)	0.53(n)	0.53(n)	0.51(n)	0.55(n)	0.51(n)	0.54(n)	0.53(n)	0.54(n)
LM5	0.52(n)	0.47(n)	NA	0.51(n)	0.51(n)	0.48(n)	0.52(n)	0.48(n)	0.51(n)	0.51(n)	0.51(n)
LM4	0.52(n)	0.47(n)	0.49(n)	NA	0.50(n)	0.47(n)	0.52(n)	0.48(n)	0.50(n)	0.50(n)	0.51(n)
LM2	0.51(n)	0.47(n)	0.49(n)	0.50(n)	NA	0.47(n)	0.51(n)	0.48(n)	0.50(n)	0.50(n)	0.50(n)
LWM4(0.5)	0.54(n)	0.49(n)	0.52(n)	0.53(n)	0.53(n)	NA	0.54(n)	0.50(n)	0.53(n)	0.53(n)	0.53(n)
LWM2(0.5)	0.50(n)	0.45(n)	0.48(n)	0.48(n)	0.49(n)	0.46(n)	NA	0.46(n)	0.49(n)	0.49(n)	0.49(n)
LWM4(0.75)	0.54(n)	0.49(n)	0.52(n)	0.52(n)	0.52(n)	0.50(n)	0.54(n)	NA	0.53(n)	0.53(n)	0.53(n)
LWM2(0.75)	0.51(n)	0.46(n)	0.49(n)	0.50(n)	0.50(n)	0.47(n)	0.51(n)	0.47(n)	NA	0.50(n)	0.50(n)
LWM4(0.95)	0.51(n)	0.47(n)	0.49(n)	0.50(n)	0.50(n)	0.47(n)	0.51(n)	0.47(n)	0.50(n)	NA	0.50(n)
LWM2(0.95)	0.51(n)	0.46(n)	0.49(n)	0.49(n)	0.50(n)	0.47(n)	0.51(n)	0.47(n)	0.50(n)	0.50(n)	NA

that, as already observed for the ISBSG dataset, HLFPA tends to underestimate. All the other models either provide estimation errors that are equally distributed between negative and positive, or (like SiFP, LWM4(0.75) and LWM4(0.95)) overestimate.

Figure 6 shows the boxplots of absolute estimation errors for each of the used methods, excluding outliers. The mean absolute error (i.e., the MAR) is shown as an orange diamond. Figure 6 shows that most models provide similar accuracy. The only models that yield evidently less accurate estimates are SiFP, LWM4(0.5) and LWM4(0.75). Concerning SiFP, it is useful reminding that it is not an estimation method, hence it is not correct to talk about estimation errors, in this case; rather, we should talk about the distance between SiFP measures and standard FPA size.

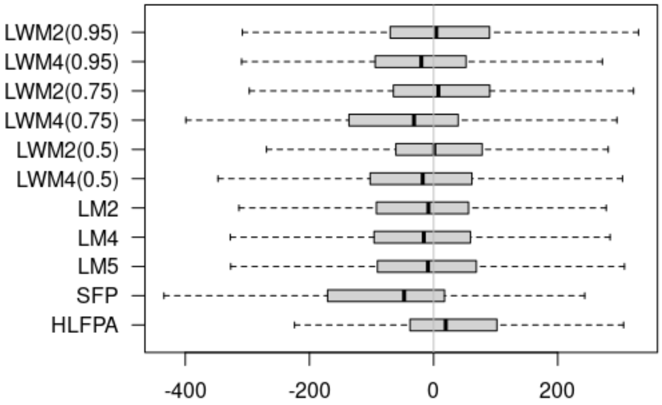


Fig. 5. Within-dataset evaluation using the Chinese dataset: error boxplots (no outliers).

With both datasets, the lowest MAR is obtained by using a LOESS approach, although with different spans. This confirms the flexibility of the method and its adaptability to different datasets after a tuning phase regarding the configuration of the span based on the peculiarities of each dataset.

F. Results of cross-dataset evaluations

This activity consisted of two steps:

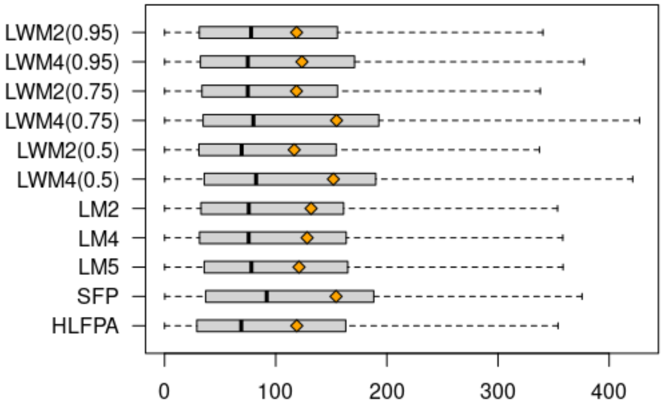


Fig. 6. Within-dataset evaluation using the Chinese dataset: absolute error boxplots (no outliers).

1) We built models using the ISBSG dataset as the training set and used the obtained model to estimate the size of projects in the Chinese dataset.
2) We built models using the Chinese dataset as the training set and used the obtained model to estimate the size of projects in the ISBSG dataset.

TABLE IX
CROSS-DATASET EVALUATION (TRAINING SET ISBSG, TEST SET CHINESE): ACCURACY INDICATORS.

	MAR	MR	MdAR	MdR	MMRE	MdMRE	R^2
HLFPA	119.0	0.088	69.0	0.066	0.095	0.077	0.970
SFP	154.3	0.114	91.9	0.088	0.124	0.108	0.945
LM5	140.7	0.104	82.8	0.080	0.112	0.088	0.955
LM4	143.1	0.106	85.3	0.082	0.113	0.090	0.953
LWM4(0.5)	403.3	0.322	147.8	0.135	0.273	0.188	0.144
LWM2(0.5)	218.1	0.148	144.5	0.114	0.146	0.123	0.859
LWM4(0.75)	315.7	0.252	129.9	0.119	0.208	0.152	0.534
LWM2(0.75)	180.2	0.122	114.2	0.090	0.119	0.107	0.920
LWM4(0.95)	241.4	0.192	106.4	0.097	0.163	0.115	0.724
LWM2(0.95)	168.5	0.114	113.5	0.090	0.113	0.100	0.929

When considering point 1) the comparison of Table VI with Table IX shows that prediction accuracy decreases for all models when "foreign" data are used for training. Of course, the accuracy obtained by HLFPA and SFP do not change, since these predictions are not obtained from any dataset.

Noticeably, models obtained via linear regression achieve a level of accuracy that is quite close to HLFPA's and slightly

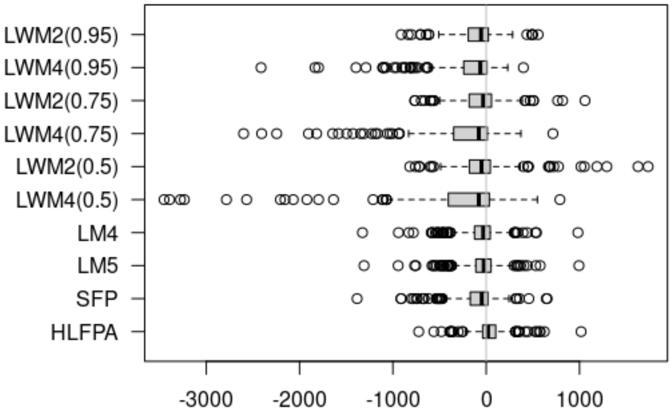


Fig. 7. Cross-dataset evaluation (training set ISBSG, test set Chinese): error boxplots.

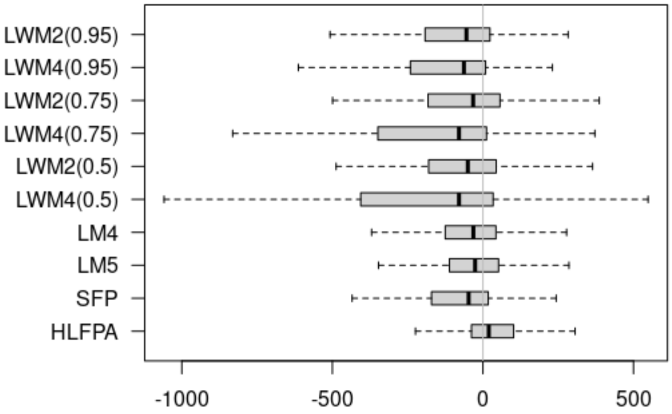


Fig. 8. Cross-dataset evaluation (training set ISBSG, test set Chinese): error boxplots (no outliers).

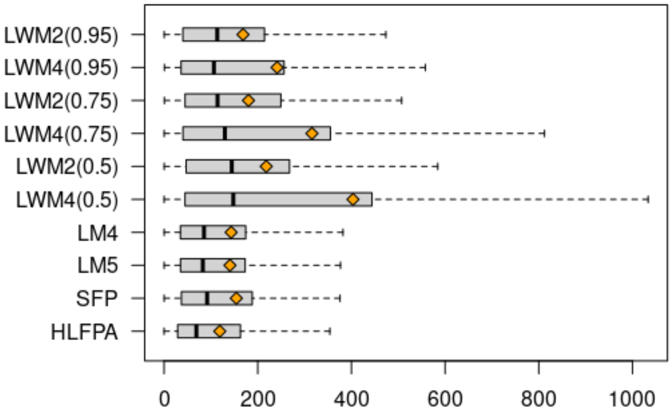


Fig. 9. Cross-dataset evaluation (training set ISBSG, test set Chinese): absolute error boxplots (no outliers).

TABLE X
CROSS-DATASET EVALUATION (TRAINING SET ISBSG, TEST SET CHINESE): COMPARISON OF MODELS' ABSOLUTE RESIDUALS VIA WILCOXON SIGN RANK TEST.

	HLFPA	SFP	LM5	LM4	LWM4	LWM2	LWM4	LWM2	LWM4	LWM2
					(0.5)	(0.5)	(0.75)	(0.75)	(0.95)	(0.95)
HLFPA	–	<	<	<	<	<	<	<	<	<
SFP	>	–	>	>	<	<	<	<	<	<
LM5	>	<	–	=	<	<	<	<	<	<
LM4	>	<	=	–	<	<	<	<	<	<
LWM4(0.5)	>	>	>	>	–	>	>	>	>	>
LWM2(0.5)	>	>	>	>	<	–	=	>	>	>
LWM4(0.75)	>	>	>	>	<	=	–	>	>	>
LWM2(0.75)	>	>	>	>	<	<	<	–	>	>
LWM4(0.95)	>	>	>	>	<	<	<	<	–	=
LWM2(0.95)	>	>	>	>	<	<	<	<	=	–

better than SFP's. However, this applies for models using 4 or 5 variables; no valid model using 2 variables could be found via linear regression. LOESS models appear definitely less accurate, although LWM2(0.95) appear only slightly less accurate than SFP.

The results of the Wilcoxon sign rank test are given in Table X. The results of the Vargha and Delaney's A test are given in Table XI.

According to the Wilcoxon sign rank test, HLFPA is the most accurate method, although according to A, the difference in accuracy is negligible when compared to SFP and linear regression models, and small when compared to LOESS models.

Figure 7 and Figure 8 show the boxplots of estimation errors for each of the used methods with and without outliers, respectively.

From both figures it can be noticed that, as already observed for the ISBSG and the Chinese dataset, HLFPA tends to underestimate. All the other models tend to overestimate, in some cases by fairly large amounts.

Figure 9 shows the boxplots of absolute estimation errors for each of the used methods, excluding outliers. It can be noticed that HLFPA, SFP and LM models provide similar and the better accuracy. All LWM4 models yield evidently less accurate estimates than LWM2.

When considering point 2) i.e., the estimation of the ISBSG dataset via models obtained from the Chinese dataset, the comparison of Table III with Table XII confirms that prediction accuracy decreases for all models when "foreign" data are used for training.

However, both linear regression and LOESS models achieve better results than HLFPA when using 4 or 5 variables. Among 2-variable models, SFP and linear regression appear more accurate than HLFPA, while LOESS models achieve slightly worse accuracy.

The results of the Wilcoxon sign rank test are given in Table XIII. The results of the Vargha and Delaney's A test are given in Table XIV.

According to the Wilcoxon sign rank test, LOESS models using 4 variables are the most accurate. According to A, LOESS models using 4 variables provide a small advantage

TABLE XI
CROSS-DATASET EVALUATION (TRAINING SET ISBSG, TEST SET CHINESE): EFFECT SIZE ACCORDING TO VARGHA AND DELANEY'S A.

	HLFPA	SFP	LM5	LM4	LWM4 (0.5)	LWM2 (0.5)	LWM4 (0.75)	LWM2 (0.75)	LWM4 (0.95)	LWM2 (0.95)
HLFPA	NA	0.45(n)	0.47(n)	0.47(n)	0.34(s)	0.36(s)	0.39(s)	0.40(s)	0.42(s)	0.42(s)
SFP	0.55(n)	NA	0.52(n)	0.51(n)	0.38(s)	0.41(s)	0.43(s)	0.44(n)	0.46(n)	0.46(n)
LM5	0.53(n)	0.48(n)	NA	0.50(n)	0.37(s)	0.39(s)	0.41(s)	0.43(n)	0.45(n)	0.45(n)
LM4	0.53(n)	0.49(n)	0.50(n)	NA	0.37(s)	0.40(s)	0.41(s)	0.43(n)	0.45(n)	0.45(n)
LWM4(0.5)	0.66(s)	0.62(s)	0.63(s)	0.63(s)	NA	0.54(n)	0.54(n)	0.57(n)	0.58(s)	0.59(s)
LWM2(0.5)	0.64(s)	0.59(s)	0.61(s)	0.60(s)	0.46(n)	NA	0.50(n)	0.53(n)	0.54(n)	0.56(n)
LWM4(0.75)	0.61(s)	0.57(s)	0.59(s)	0.59(s)	0.46(n)	0.50(n)	NA	0.53(n)	0.54(n)	0.54(n)
LWM2(0.75)	0.60(s)	0.56(n)	0.57(n)	0.57(n)	0.43(n)	0.47(n)	0.47(n)	NA	0.51(n)	0.52(n)
LWM4(0.95)	0.58(s)	0.54(n)	0.55(n)	0.55(n)	0.42(s)	0.46(n)	0.46(n)	0.49(n)	NA	0.50(n)
LWM2(0.95)	0.58(s)	0.54(n)	0.55(n)	0.55(n)	0.41(s)	0.44(n)	0.46(n)	0.48(n)	0.50(n)	NA

over HLFPA and SFP, while the advantage is negligible with respect to linear regression models.

Figure 10 and Figure 11 show the boxplots of estimation errors for each of the used methods, with and without outliers, respectively. The boxplots show that most methods tend to underestimate. LWM4 models are either well balanced or tend to overestimate. Similarly, SFP tends to overestimate.

Figure 12 shows the boxplots of absolute estimation errors for each of the used methods, excluding outliers. It can be noticed that the better accuracy is provided by LMW4 methods, while HLFPA, LM2 and all the LMW2 provide similar and worse accuracy with respect to the other methods. LM methods are between those extremes.

TABLE XII
CROSS-DATASET EVALUATION (TRAINING SET CHINESE, TEST SET ISBSG): ACCURACY INDICATORS.

	MAR	MR	MdAR	MdR	MMRE	MdMRE	R2
HLFPA	103.8	0.106	58.0	0.091	0.097	0.084	0.966
SFP	87.1	0.089	60.5	0.095	0.105	0.078	0.978
LM5	90.3	0.093	51.8	0.081	0.090	0.083	0.976
LM4	81.9	0.084	48.5	0.076	0.086	0.080	0.978
LM2	108.0	0.111	57.8	0.091	0.106	0.100	0.959
LWM4(0.5)	63.6	0.069	35.5	0.057	0.075	0.058	0.980
LWM2(0.5)	115.1	0.118	62.7	0.098	0.107	0.094	0.947
LWM4(0.75)	69.8	0.076	50.8	0.082	0.082	0.063	0.979
LWM2(0.75)	118.5	0.121	65.2	0.102	0.108	0.088	0.941
LWM4(0.95)	66.9	0.073	42.9	0.069	0.077	0.059	0.978
LWM2(0.95)	113.7	0.117	58.4	0.091	0.102	0.087	0.945

TABLE XIII
CROSS-DATASET EVALUATION (TRAINING SET CHINESE, TEST SET ISBSG): COMPARISON OF MODELS' ABSOLUTE RESIDUALS VIA WILCOXON SIGN RANK TEST.

	HLFPA	SFP	LM5	LM4	LWM4 (0.5)	LWM2 (0.5)	LWM4 (0.75)	LWM2 (0.75)	LWM4 (0.95)	LWM2 (0.95)	
HLFPA	–	>	>	>	=	>	=	>	=	>	>
SFP	<	–	=	>	<	>	=	>	=	>	=
LM5	<	=	–	>	<	>	<	>	=	>	=
LM4	<	<	<	–	<	>	<	>	<	>	<
LM2	=	>	>	>	–	>	=	>	=	>	>
LWM4(0.5)	<	<	<	<	<	–	<	<	<	=	<
LWM2(0.5)	=	=	>	>	=	>	–	>	=	>	=
LWM4(0.75)	<	<	<	<	<	>	<	–	<	>	<
LWM2(0.75)	=	=	=	>	=	>	=	>	–	>	=
LWM4(0.95)	<	<	<	<	<	=	<	<	<	–	<
LWM2(0.95)	<	=	=	>	<	>	=	>	=	>	–

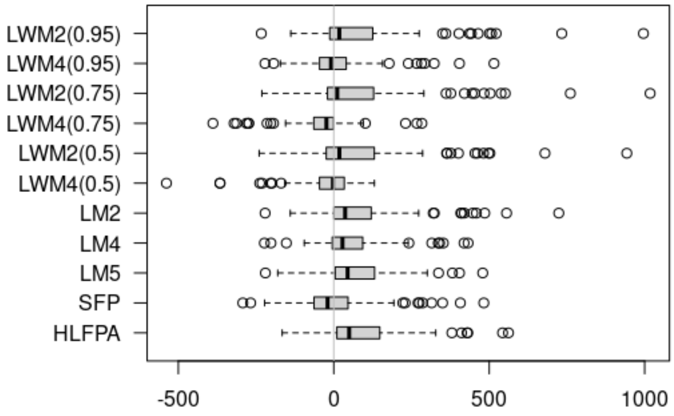


Fig. 10. Cross-dataset evaluation (training set Chinese, test set ISBSG): error boxplots.

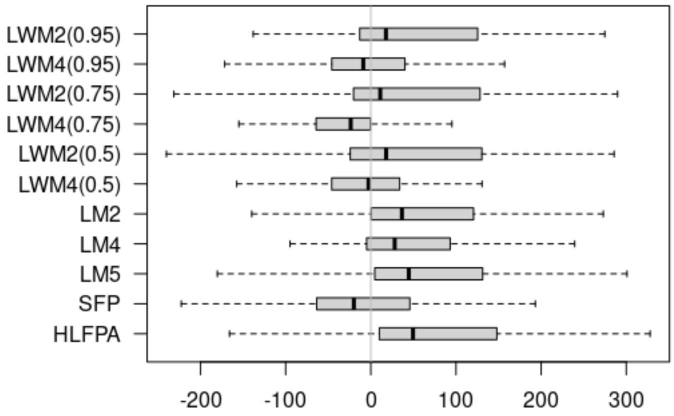


Fig. 11. Cross-dataset evaluation (training set Chinese, test set ISBSG): error boxplots (no outliers).

IV. DISCUSSION

In this section we discuss the obtained results from two points of view: a technical one (in Section IV-A) and a managerial one (in Section IV-B).

TABLE XIV
CROSS-DATASET EVALUATION (TRAINING SET CHINESE, TEST SET ISBSG): EFFECT SIZE ACCORDING TO VARGHA AND DELANEY'S A.

	HLFPA	SFP	LM5	LM4	LM2	LWM4	LWM2	LWM4	LWM2	LWM4	LWM2
						(0.5)	(0.5)	(0.75)	(0.75)	(0.95)	(0.95)
HLFPA	NA	0.52(n)	0.52(n)	0.55(n)	0.50(n)	0.61(s)	0.51(n)	0.59(s)	0.51(n)	0.60(s)	0.53(n)
SFP	0.48(n)	NA	0.51(n)	0.54(n)	0.48(n)	0.60(s)	0.49(n)	0.58(s)	0.49(n)	0.59(s)	0.51(n)
LM5	0.48(n)	0.49(n)	NA	0.53(n)	0.47(n)	0.58(s)	0.48(n)	0.56(n)	0.49(n)	0.58(s)	0.50(n)
LM4	0.45(n)	0.46(n)	0.47(n)	NA	0.45(n)	0.56(n)	0.45(n)	0.53(n)	0.45(n)	0.55(n)	0.47(n)
LM2	0.50(n)	0.52(n)	0.53(n)	0.55(n)	NA	0.61(s)	0.50(n)	0.58(s)	0.51(n)	0.60(s)	0.52(n)
LWM4(0.5)	0.39(s)	0.40(s)	0.42(s)	0.44(n)	0.39(s)	NA	0.40(s)	0.47(n)	0.41(s)	0.49(n)	0.42(s)
LWM2(0.5)	0.49(n)	0.51(n)	0.52(n)	0.55(n)	0.50(n)	0.60(s)	NA	0.58(s)	0.50(n)	0.59(s)	0.51(n)
LWM4(0.75)	0.41(s)	0.42(s)	0.44(n)	0.47(n)	0.42(s)	0.53(n)	0.42(s)	NA	0.43(s)	0.52(n)	0.44(n)
LWM2(0.75)	0.49(n)	0.51(n)	0.51(n)	0.55(n)	0.49(n)	0.59(s)	0.50(n)	0.57(s)	NA	0.59(s)	0.52(n)
LWM4(0.95)	0.40(s)	0.41(s)	0.42(s)	0.45(n)	0.40(s)	0.51(n)	0.41(s)	0.48(n)	0.41(s)	NA	0.43(n)
LWM2(0.95)	0.47(n)	0.49(n)	0.50(n)	0.53(n)	0.48(n)	0.58(s)	0.49(n)	0.56(n)	0.48(n)	0.57(n)	NA

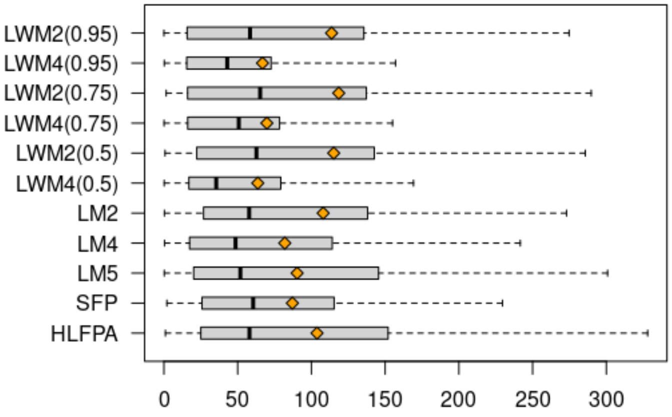


Fig. 12. Cross-dataset evaluation (training set Chinese, test set ISBSG): absolute error boxplots (no outliers).

A. Technical discussion

The approaches to size estimation presented in the previous sections correspond to different model building strategies, which are based on different assumptions and require different types of knowledge. In fact,

- HLFPA exploits the knowledge of how FPA works. According to FPA, the measure of size is obtained as a weighted sum of the numbers of EI, EO, EQ, ILF and EIF. HLFPA adopts exactly the same schema. HLFPA does not rely on any data, i.e., the model is fixed and does not depend on the characteristics of the known projects. In other words, HLFPA does not try to learn from data; instead, it simply adopts fixed weights, namely low complexity weights for data and medium complexity weights for transactions.
- SFP works along similar lines. Structurally, it is a simplified version of FPA. Like HLFPA, it does not learn from data, i.e., it does not try to adapt to the characteristics of the known projects. Even though the weights to be used were originally derived by the observation of data from real projects, these weights are now fixed and apply to whatever project has to be measured.
- OLS exploit, like HLFPA, the knowledge of the structure of FPA sizing, in that they model size as a linear function of the numbers of EI, EO, EQ, ILF and EIF. In addition, OLS linear regression models also exploit data from known projects, since they derive the weights to be used in the computation of size from historical data. Accordingly, any organization owing suitable historical data can build its own OLS model.
- LOESS is a more flexible method with respect to OLS in that it builds models based on ML approaches (like nearest neighbours), also keeping the simplicity of regression models. Using locality principles, it may possibly yield more accurate estimates than OLS methods.

The size of the dataset may hinder the performance of the LOESS method. As a counterpart, in cross-dataset validation, LOESS models showed the best performances of the whole set of experiments. This may suggest that the generalizability of this approach should be further analyzed in search of specific conditions for a better performance of the algorithm.

B. Managerial Discussion

From the managerial standpoint, LOESS has some limitations and potential, depending on its use and application context.

With respect to HLFPA and SFP, the LOESS-based methods have the disadvantange that they need to be trained on a dataset, while the former models are fixed formulae (see (1) and (2)) that just need measures from the project being estimated. Therefore, an historical dataset is needed, and using "foreign" data may not work well, as in the case of the models trained on the ISBSG dataset and use to estimate projects from the Chinese dataset. However, using LOESS models yielded quite accurate estimates in several cases, therefore it is seems that build LOESS models is worth trying, when data are available for training. In this respect, the work needed to build LOESS models is similar to the work needed to build linear regression models, which is a fairly common activity.

It must also be considered that in some contexts, like public sectors, for instance, estimates base on LOESS models may be difficult to accept, depending on the kind of contractors. An estimation tool like LOESS could seem not transparent enough to yield reliable estimated to be agreed upon.

However, in general, from the organizational and managerial perspectives, using the LOESS method could be useful for the early assessment of the feasibility of a project before any elicitation phase, as in the case of agile methodologies. Pursuing the study of functional size estimation via LOESS may act as a proof of concept mechanism to help identify project features; to simulate and quantify the average error and intrinsic residuality of early estimation methods vs post-hoc measurement; to help compare functional size models and estimation procedures and their measurement validity and reliability.

A further opportunity represented by this approach is that of introducing evolutionary-wise estimation methods, whereby different outcomes may come from the identification of the same BFC (and whereby, in this respect, fixed weights methods would always return the same outcome). In this light, LOESS may represent a more situated approach, evolving through time and in line with factors characterizing and influencing from time to time the productive system.

C. Applicability

In this section, the practical applicability of LOESS for functional size estimation is briefly discussed.

First of all, to use LOESS, we need historical data. Besides the usual requirements for data, we need data that represent the entire size range in which we are interested. Specifically, LOESS requires fairly large, densely sampled data sets in order to produce good models. Remember that LOESS performs local fitting, therefore fairly complete information concerning BFC configurations have to be available.

Besides data, we just need a reasonable computer environment. A modern PC running the R environment (the `loess` function is available by default).

As we reported above, LOESS works well, but does not always provide the best estimates. Therefore, we do not recommend replacing estimation practices based on HLFPA or linear regression models, for instance, with LOESS right away. Instead, it can be useful to use LOESS alongside other estimation methods. In this way, if LOESS results agree with other methods' estimates you increase your confidence in the correctness of estimates. Otherwise, i.e., if LOESS disagrees with other methods' estimate, you should regard all the obtained estimates as subject to some uncertainty.

V. Threats to validity

A typical concern in this kind of studies is the generalizability of results outside the scope and context of the analyzed dataset. In our case, the ISBSG dataset is deemed the standard benchmark among the community, and it includes data from several application domains. Therefore our results may be valid in general. However, this dataset resulted too small for local approaches like LOESS, which showed its effectiveness and efficiency when applied to a larger dataset as the Chinese one. This may also suggest a limitation of the approach related to the specific dataset that each time is used. For this reason, the problem of generalizability remains crucial.

The usage of MMRE is questionable, since it is has been shown to be a biased indicator (see for instance [19]). Nonetheless, we used MMRE together with other indicators—like MAR, the boxplots of residuals and R^2—to provide a more complete and balanced picture of the accuracy of our results, and compared the precision of different models via sound statistical tests, namely Wilcoxon sign rank test and Vargha and Delaney's A measure of effect size. Therefore, the role of MMRE in the presented evaluations is marginal. Although the comparison of precision did not always yield significant differences, it is nonetheless a formal and robust method for comparing the used techniques.

VI. Related work

The quest for measures that are available in the early stages of the software lifecycle dates back to decades ago [37] [38] [30].

The "Early & Quick Function Point" (EQFP) method [32] uses analogy (similarities between a new and a classified piece of software) and analysis (statistical analysis of the estimated similarity) to get size estimates. It was reported that estimates are within $\pm 10\%$ of the real size in most real cases, while the savings in time and costs are between 50% and 90%.

"Easy Function Points," [39], adopt probabilistic approaches to estimate not only the size, but also the probability that the actual size is equal to the estimate.

Lavazza et al. built estimation models for UFP based on BFCs [40] using Least Median Squares robust regression models. They observed that FP measures could be altogether replaced by measured based on a smaller set of BFCs.

Several other early estimation methods were proposed: Table XV list the most popular ones.

Lavazza and Liu [11] used 7 real-time applications and 6 non real-time applications to evaluate the accuracy of the E&QFP [30] and HLFPA methods with respect to full-fledged Function Point Analysis. The results showed that the Indicative FPA method yields the greatest errors. On the contrary, the HLFPA method yields size estimates that are close to the actual size. Specifically, the HLFPA method proved fairly good in estimating both Real-Time and non Real-Time applications.

Lavazza and Liu [16] used a dataset containing data from 479 projects to compare the accuracy of HLFPA method with Ordinary Least Squares method, with both 5 predictors (LM5) and only 2 predictors (LM2). Their conclusions were that, although HLFPA method is sufficiently accurate for practical usage, it tends to underestimate effort. Since underestimation may lead to unrealistic development plans and possibly to project failure, the authors looked for motivations of HLFPA method underestimation behaviour, finding that it assumes that data functions are mainly of low complexity and transaction functions are mainly of medium complexity, while in the considered dataset it was not so. An alternative strategy they derived from it is to compute linear regression in order to derive the most likely weight by analyzing the data from projects. They found that (1) unlike HLFPA, linear regression models do not underestimate, (2) linear regression models yield slightly less accurate estimates, and (3) models based on only two variables yield marginally less accurate estimates.

TABLE XV
EARLY ESTIMATION METHODS: DEFINITIONS AND EVALUATIONS

Method name	Definition	Used functions	Weight	Evaluation
NESMA indicative	[24] [25]	data	fixed	[5] [15], [26]–[29] [11]
NESMA estimated	[24] [25]	all functions	fixed	[5] [15], [26]–[29] [11]
Early & Quick FP	[30] [31] [32]	all functions	statistics	[11] [33]
Tichenor ILF model	[34]	ILF	fixed	[11]
simplified FP (sFP)	[35]	all functions	fixed	[11]
ISBSG average weights	[36]	all functions	statistics	[11]
SiFP	[6]	data and trans.	statistics	[12] [13]

Also Machine learning (ML) techniques have proved to provide quite good estimation models, in several different domains and situations, and are increasingly being used in software project management activities [41], [42]. A review of the usage of ML for software project management [42] reported that ML is used for software effort and cost estimation: the reported accuracy spans from 91% for cost estimation with K-NN (K-Nearest Neighbours), to 92% for effort prediction with Decision Trees, and 99% for effort estimation with Random Forests.

Local regression methods are extensively used for DNA microarray normalization studies [43], as well as for studying spatiotemporal trends, and improving image resolution and forecasts predictions. They have also been used for hand tracking rapid movements in Human-Computer Interaction studies [44]. However, regarding software size estimation, only a few study have focused on the use of LOESS (see for example [45]), by comparing this method with other ML approaches. In this paper, we are interested in the estimation of functional size, which is generally the main input for effort estimation. Approaches based on local regression have been rarely adopted in this field. We hope to have contributed in a constructive way to better introduce this technique for the analysis and the modelling of software functional size.

VII. CONCLUSION

Measuring software functional size via IFPUG FPA with the standard manual process is sometimes a long and expensive activity, and it is simply impossible when the details of a functional specification are not available for any reason. To solve this problem, several early estimation methods have been proposed. In this paper, we compare the estimates obtained via a standard estimation methods, namely HLFPA, and a new functional size measurement method, namely IFPUG SFP, with the estimates obtained with traditional (namely, linear regression) models and LOESS models.

To evaluate the accuracy of the functional size estimates provided by the considered methods, we performed both within-dataset and cross-dataset studies. Specifically, we performed two within-datasets analyses, one using an ISBSG dataset containing data from 110 projects and one using a dataset containing data from 276 software projects developed and used by a Chinese financial enterprise. We then performed two cross-datasets analysis: in the first one the ISBSG dataset was used for training and the Chinese dataset was used for testing; in the second one the Chinese dataset was used for training and the ISBSG dataset was used for testing.

When performing within-dataset evaluation using the ISBSG dataset, the LOESS and linear regression models provided the best MAR. Among models using only two variables (unclassified data and transaction functions) the LOESS and SFP models provided the best MAR.

When performing within-dataset evaluation using the Chinese dataset, HLFPA provided the best MAR, with the linear regression and LOESS models providing very similar performance. Among models using only two variables the LOESS model provided the best results, even better than HLFPA's.

When using the ISBSG dataset to train models and the Chinese dataset for testing, HLFPA was definitely most accurate than other models. However, when using the Chinese dataset to train models and the ISBSG dataset for testing, the LOESS model provided definitely the best results. SFP proved also quite good.

We assessed the effect size via the non-parametric statistic A by Vargha and Delaney; we also compared the absolute residuals via Wilcoxon sign rank test to evaluate if the estimates provided by a method are significantly better than those provided by another method. In general, the obtained results show that no methods appears consistently better than others, and the differences are small or even negligible.

In conclusion, even though there is no clear winner, the LOESS method provided generally quite good results; therefore, practitioners needing to estimate software functional size in the early stages of projects are advised to try also LOESS models.

Among future work, we envision the following activities:

- Comparing LOESS estimates with those produced by machine learning techniques [46].
- Study LOESS estimates with confidence intervals.
- Evaluating size estimates obtained via LOESS models, when used for effort estimation.

ACKNOWLEDGMENT

The work reported here was partly supported by Fondo per la Ricerca di Ateneo, Università degli Studi dell'Insubria.

REFERENCES

[1] L. Lavazza, A. Locoro, and R. Meli, "Using Locally Weighted Regression to Estimate the Functional Size of Software: a Preliminary Study," in Proceedings of IARIA Congress 2022: The 2022 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications, 2022, pp. 20–24.

[2] A. J. Albrecht, “Measuring application development productivity,” in Proceedings of the joint SHARE/GUIDE/IBM application development symposium, vol. 10, 1979, pp. 83–92.

[3] International Function Point Users Group (IFPUG), “Function point counting practices manual, release 4.3.1,” 2010.

[4] A. Timp, “uTip – Early Function Point Analysis and Consistent Cost Estimating,” 2015, uTip # 03 – (version # 1.0 2015/07/01).

[5] H. van Heeringen, E. van Gorp, and T. Prins, “Functional size measurement-accuracy versus costs–is it really worth it?” in Software Measurement European Forum (SMEF), 2009.

[6] R. Meli, “Simple function point: a new functional size measurement method fully compliant with IFPUG 4.x,” in Software Measurement European Forum, 2011.

[7] IFPUG, “Simple Function Point (SFP) Counting Practices Manual Release 2.1,” 2021.

[8] International Standardization Organization (ISO), “ISO/IEC 20926: 2003, Software engineering – IFPUG 4.1 Unadjusted functional size measurement method – Counting Practices Manual,” 2003.

[9] L. Lavazza, “On the effort required by function point measurement phases,” International Journal on Advances in Software, vol. 10, no. 1 & 2, 2017.

[10] nesma, “Early Function Point Analysis,” https://nesma.org/themes/sizing/function-point-analysis/early-function-point-counting/ last access 6/6/22.

[11] L. Lavazza and G. Liu, “An empirical evaluation of simplified function point measurement processes,” Journal on Advances in Software, vol. 6, no. 1& 2, 2013.

[12] L. Lavazza and R. Meli, “An evaluation of simple function point as a replacement of IFPUG function point,” in IWSM–MENSURA 2014. IEEE, 2014, pp. 196–206.

[13] F. Ferrucci, C. Gravino, and L. Lavazza, “Simple function points for effort estimation: a further assessment,” in 31st Annual ACM Symposium on Applied Computing. ACM, 2016, pp. 1428–1433.

[14] International Software Benchmarking Standards Group, ““Worldwide Software Development: The Benchmark, release 11,” ISBSG, 2009.

[15] L. Lavazza and G. Liu, “An Empirical Evaluation of the Accuracy of NESMA Function Points Estimates,” in ICSEA, 2019, pp. 24–29.

[16] G. Liu and L. Lavazza, “Early and quick function points analysis: Evaluations and proposals,” Journal of Systems and Software, vol. 174, 2021, p. 110888.

[17] W. S. Cleveland, “Robust locally weighted regression and smoothing scatterplots,” Journal of the American statistical association, vol. 74, no. 368, 1979, pp. 829–836.

[18] R core team, “R: a language and environment for statistical computing,” 2015.

[19] B. Kitchenham, L. Pickard, S. MacDonell, and M. Shepperd, “What accuracy statistics really measure [software estimation],” in Software, IEE Proceedings-, vol. 148, no. 3. IET, 2001, pp. 81–85.

[20] D. Chicco, M. J. Warrens, and G. Jurman, “The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation,” PeerJ Computer Science, vol. 7, 2021, p. e623.

[21] A. Vargha and H. D. Delaney, “A critique and improvement of the cl common language effect size statistics of mcgraw and wong,” Journal of Educational and Behavioral Statistics, vol. 25, no. 2, 2000, pp. 101–132.

[22] M. Torchiano et al., “effsize: Efficient effect size computation,” R package version 0.7, vol. 1, 2017.

[23] J. Cohen, “Statistical power analysis for the behavioral sciences Lawrence Earlbaum Associates,” Hillsdale, NJ, 1988, pp. 20–26.

[24] NESMA–the Netherlands Software Metrics Association, “Definitions and counting guidelines for the application of function point analysis. NESMA Functional Size Measurement method compliant to ISO/IEC 24570 version 2.1,” 2004.

[25] International Standards Organisation, “ISO/IEC 24570:2005 – Software Engineering – NESMA functional size measurement method version 2.1 – definitions and counting guidelines for the application of Function Point Analysis,” 2005.

[26] F. G. Wilkie, I. R. McChesney, P. Morrow, C. Tuxworth, and N. Lester, “The value of software sizing,” Information and Software Technology, vol. 53, no. 11, 2011, pp. 1236–1249.

[27] J. Popović and D. Bojić, “A comparative evaluation of effort estimation methods in the software life cycle,” Computer Science and Information Systems, vol. 9, no. 1, 2012, pp. 455–484.

[28] P. Morrow, F. G. Wilkie, and I. McChesney, “Function point analysis using nesma: simplifying the sizing without simplifying the size,” Software Quality Journal, vol. 22, no. 4, 2014, pp. 611–660.

[29] S. Di Martino, F. Ferrucci, C. Gravino, and F. Sarro, “Assessing the effectiveness of approximate functional sizing approaches for effort estimation,” Information and Software Technology, vol. 123, July 2020.

[30] L. Santillo, M. Conte, and R. Meli, “Early & Quick Function Point: sizing more with less,” in 11th IEEE International Software Metrics Symposium (METRICS’05). IEEE, 2005, pp. 41–41.

[31] T. Iorio, R. Meli, and F. Perna, “Early&quick function points® v3. 0: enhancements for a publicly available method,” in SMEF, 2007, pp. 179–198.

[32] DPO, “Early & Quick Function Points Reference Manual - IFPUG version,” DPO, Roma, Italy, Tech. Rep. EQ&FP-IFPUG-31-RM-11-EN-P, April 2012.

[33] R. Meli, “Early & quick function point method-an empirical validation experiment,” in Int. Conf. on Advances and Trends in Software Engineering, Barcelona, Spain, 2015.

[34] C. Tichenor, “The IRS development and application of the internal logical file model to estimate function point counts,” in IFPUG Fall Conf., 1997.

[35] L. Bernstein and C. M. Yuhas, Trustworthy systems through quantitative software engineering. John Wiley & Sons, 2005, vol. 1.

[36] R. Meli and L. Santillo, “Function point estimation methods: A comparative overview,” in FESMA, vol. 99. Citeseer, 1999, pp. 6–8.

[37] D. B. Bock and R. Klepper, “FP-S: a simplified function point counting method,” Journal of Systems and Software, vol. 18, no. 3, 1992, pp. 245–254.

[38] G. Horgan, S. Khaddaj, and P. Forte, “Construction of an FPA-type metric for early lifecycle estimation,” Information and Software Technology, vol. 40, no. 8, 1998, pp. 409–415.

[39] L. Santillo, “Easy Function Points – ‘Smart’ Approximation Technique for the IFPUG and COSMIC Methods,” in IWSM–MENSURA, 2012.

[40] L. Lavazza, S. Morasca, and G. Robiolo, “Towards a simplified definition of function points,” Information and Software Technology, vol. 55, no. 10, 2013, pp. 1796–1809.

[41] P. Pospieszny, B. Czarnacka-Chrobot, and A. Kobylinski, “An effective approach for software project effort and duration estimation with machine learning algorithms,” Journal of Systems and Software, vol. 137, 2018, pp. 184–196.

[42] M. N. Mahdi, M. H. Mohamed Zabil, A. R. Ahmad, R. Ismail, Y. Yusoff, L. K. Cheng, M. S. B. M. Azmi, H. Natiq, and H. Happala Naidu, “Software project management using machine learning technique—a review,” Applied Sciences, vol. 11, no. 11, 2021, p. 5183.

[43] X. Liu, N. Li, S. Liu, J. Wang, N. Zhang, X. Zheng, K.-S. Leung, and L. Cheng, “Normalization methods for the analysis of unbalanced transcriptome data: a review,” Frontiers in bioengineering and biotechnology, vol. 7, 2019, p. 358.

[44] T. Kuronen, T. Eerola, L. Lensu, J. Takatalo, J. Häkkinen, and H. Kälviäinen, “High-speed hand tracking for studying human-computer interaction,” in Scandinavian Conference on Image Analysis. Springer, 2015, pp. 130–141.

[45] L. Q. Leal, R. A. Fagundes, R. M. de Souza, H. P. Moura, and C. M. Gusmão, “Nearest-neighborhood linear regression in an application with software effort estimation,” in 2009 IEEE International Conference on Systems, Man and Cybernetics. IEEE, 2009, pp. 5030–5034.

[46] L. Lavazza, A. Locoro, G. Liu, and R. Meli, “Estimating software functional size via machine learning,” ACM Transaction on Software Engineering and Methodology, vol. to appear, no. ?, 2023?, p. ?

Database Technology Evolution

Malcolm Crowe
Emeritus Professor, Computing Science
University of the West of Scotland
Paisley, United Kingdom
Malcolm.Crowe@uws.ac.uk

Fritz Laux
Emeritus Professor, Business Computing
Universität Reutlingen
Reutlingen, Germany
Fritz.Laux@reutlingen-university.de

Abstract– **This paper reviews suggestions for changes to database technology coming from the work of many researchers, particularly those working with evolving big data. We discuss new approaches to remote data access and standards that better provide for durability and auditability in settings including business and scientific computing. We propose ways in which the language standards could evolve, with proof-of-concept implementations on Github.**

Keywords– big live data; remote data; RDBMS; SQL; standards.

I. Introduction

The design of relational database management systems (RDBMS) has always focused on the management of structured and evolving data, such as customer accounts and scientific results, where shared access and long-term durability are important [1]. The Standard Query Language SQL, developed in the 1970s, rapidly became an international standard [2] with many features, and its evolution has been followed by most database products. Many researchers have been inspired to develop the theoretical underpinning for the implementation of these products, and this work continues today [3][4][5][6].

With all forms of evolution, some inherited aspects become awkward over time, for example, the early use of fixed-size fields and limited precision primitive types persists in database storage, limiting backwards compatibility of newer product versions and affecting durability and portability [7]. Some research projects including PyrrhoDB have chosen instead to use new globalized primitive types to avoid dependency on machine architecture and locale [8]. Avoiding such dependency facilitates data import and sharing, and the construction of data warehouses [9].

The development of data warehouses has led to a focus on metadata and semantics and has led many systems to use document-based NoSQL systems while other researchers have developed ways of including semantics in relational systems [10]. With these developments, it is natural to seek ways of adapting the relational DBMS paradigm to manage evolving data warehouse content (big live data) [11].

The tension between evolution and durability of *data* has always been a feature of relational database management systems (RDMS) and the associated technology. The use cases that inspired RDBMS development were business records such as customer accounts and inventories, and collaborative science, where support for shared access by many users with the responsibility for keeping data up to date needs to be balanced by the requirements for long-term storage, consistency, and audit. Over the years, such support has evolved, by the addition of powerful declarative and processing features in the evolving standard language SQL [2], and this evolution has come with a cost in compatibility between systems, since not all RDBMS implement the same version of the standard, and in durability, since RDBMS products also evolve, and not all RDBMS provide adequate backward compatibility to work with databases developed for a previous version. For these reasons, legacy data and systems are a continuing concern in all forms of business and scientific endeavor.

The starting point in this contribution is that the DBMS should generally support enterprise data integration where appropriate, and co-operative data sharing where this is useful. That is, the DBMS itself should support, but not require, ways of extending a data model through the enterprise, while providing mechanisms for supporting useful applications for the situations where the responsibility for data evolution is in another organization. In both cases the resulting structure will be a federation allowing some local management, with a hierarchy of delegation and responsibility, to avoid over-centralization on the one hand, or wasteful duplication on the other. This paper considers a number of improvements to DBMS technology designed to achieve this aim, while maintaining strong safeguards for preserving consistency for such complex systems where shared data evolves through supported activities in all parts of the system.

In the next section we consider an important set of use cases where people are interested in very targeted real-time data, gathered from many sources, where queries often lead to a unique entity on a single server. SQL remains a popular way of implementing database applications and even more general query systems, and ideally any changes should remain close to its original intent. In later sections of this paper, we examine some novel open-source approaches to such use cases in the PyrrhoDB project, which are based in widely used technologies and so have the potential to be useful in future big data developments. PyrrhoDB itself is a research project dating from before 2005 [12] rather than a product, but from its beginnings it has used globalized and machine-independent structures and the international standards and has always supported both evolution and backward compatibility.

In Section II we consider the state of the art, with an analysis of recent research papers that draw attention to changing requirements in database support for large and evolving data sets. This section also creates an agenda for the rest of the paper, to consider and suggest changes to relational data technology: *serialized* transactions and hierarchical privileges in Section III, proposals for the data type system and *metadata* in Section IV, *virtual* data warehousing (view-mediated remote access) in Section V, a suggestion to build implementations using *shareable* data structures in Section VI. Section VII looks at the implications for query processing, and Section VIII proposes a *versioned* API alongside the usual SQL data access methods and compares them with those of other database products. These sections include examples, and proof-of-concept implementations of these ideas are offered on Github.

II. THE USE CASE OF BIG LIVE DATA

Raw scientific and administrative data are often meaningless to the general public but is usually carried on the public web and usually has a significant real-time aspect.

Examples:

- The DNA signature of the latest Covid variants (whose data is progressively refined) [13],
- the latest data from sensors mapping a tsunami [14],
- the treatment history of a patient with a serious illness [15],
- the results from a particular fluid calculation that has taken a supercomputer three days to compute [16],
- the history of a piece of steel reinforcement in a tower block [17],
- the availability of intensive-care equipment for an emergency hospital admission [18],
- a particular sensor or actuator in the Internet of Things [19].

In some cases, there may be expectations coming from modeling (or AI) but a lot of important people in WHO, NASA, etc. want the scientists or investigators to get the right data. In some cases, the data (e.g., from sensors) is real-time, in others (e.g., the supercomputer example) the results may be a high-resolution image from numerical results that might not even be stored anywhere. Often such requests have life-and-death implications, and in order to guard against receiving approximate or out-of-date information, people resort to email or telephone.

In all such cases the data is conceptually part of a giant sparse database that no-one could possibly construct. Any individual observations would have lots of dependent metadata (provenance, device-specific details, confidence etc.). But often, the questions that the scientists want to ask are phrased in database terms, e.g., to examine the outcomes of patients with rare diseases and specific treatments, the quality of steel used in a component that needs to be replaced etc.

If SQL querying and secure remote update is also considered desirable, the above use cases point to some potentially desirable features. Excluding already-standard aspects such as authorization, universal time, international standards, auditing and linked data, and including features that not everyone would require, we can easily come up with the following wish list for SQL support:

- Search current data from a named collection of remote data sets
- Allow searching by metadata such as the resource description framework (RDF) or provenance where available
- Ensure transmitted data comes with timed provenance and ownership information
- Ensure remote updates (if permitted) are directly handled by the data owner, and fully recorded with user information of sender
- Avoid second-hand or out-of-date data by directly accessing the data's "transaction master"
- Specify service quality. e.g., to prioritize correctness over availability, report on out-of-date data or servers offline
- Minimize the amount of data that needs to be obtained or preloaded from remote servers
- Allow for transformation during retrieval, with inverses for updates if permitted
- Ensure changes are securely transacted, and durably recorded.

From the above discussion, in what follows we are motivated by the following general considerations:

- A focus on the need to support legacy data should motivate the separation of durable data from volatile data. The current state of any individual account or evolving record needs to be accessible from memory, but as in archiving, durable systems should prioritize and enable auditing of primary data such as particular inputs, changes, and deletions. In what follows, we reserve the concept of durable storage for this archive.
- On the other hand, access to and modification of the shared state of evolving data needs effective transaction control. The capturing of the desired durable archive then amounts to a log of such transactions, and the best way to prove the serializability of recorded transactions is that this log should itself record them atomically, in commit order, with all the steps of each commit kept together. Implementation of this log should be as append storage [20]. We note that some widely-supported DBMS features such as constraints, cascades and triggers complicate this requirement.
- Most DBMS are wary of the use of the Internet and prefer managing all network interaction using custom features. In our view this is now a mistake and ignores the opportunities for globalization that the evolving Internet standards offer. Greater opportunities for access should be balanced by better recording of data ownership, provenance, and responsibility, and these would help to address the concerns noted above for the ability in special cases to obtain results from (or even to update) sources rather than copies. We will demonstrate that such increased use of Internet standards has the potential to reduce wasteful data replication, especially for "live" data.

In considering the requirements for DBMS evolution, therefore, we consider the following aspects:

- The validation of transaction serialization, taking account of all side effects of transactions, so that transactions that violate constraints should not commit, nor if a resulting cascade or triggered action will conflict with other transactions. This requirement is mandated by the international SQL standard [2] but rarely implemented in commercial DBMS.
- We suggest a modified approach to DBMS design and security that places the data model and security model in the database rather than in applications. The SQL standard provides almost all of the support needed to achieve this: we take this forward by highlighting the definer's role for precompiled code and constraints, and through the creation of metadata features for the database itself. There are some consequential suggestions for enhancing SQL's extensive data type system.
- As in the US Department of Defense Orange Book standards for mandatory access control, we place the focus on user responsibility and security, while granting permissions to roles rather than users. Our proof-of-concept code includes the features required to implement the Orange Book levels B and C for users and database objects. Roles offer privileges on objects, and users are granted roles. We suggest however that the SQL standard should be modified so that a user can only use one role at a time. This is a practical suggestion since a user can be allowed to substitute for a sick colleague, but all actions are recorded in a way that identifies both the user and their declared role at the time.
- The SQL programming model is computationally complete: we recommend that the use of external code and procedures is disallowed, so that the DBMS can manage all of the validation and auditing required.
- In these circumstances, we support ways to allow better remote access to databases in SQL.

The remaining sections of this paper deal with practical proposals for all these aspects, making minimal changes to the SQL standard. Proof-of-concept code for these ideas already exists in PyrrhoDB on Github. Details are provided here in the following feature groupings: serialized transactions, DBMS accountability and data ownership, metadata, and view-mediated remote access.

III. Serialized Transactions

From the above discussion, we implement a validation step for all transaction commits, to ensure that the requirement for fully serialized transactions is met. This renders obsolete the list of isolation levels (`READ_UNCOMMITTED`, `READ_COMMITTED`, `REPEATABLE_READ`, `SERIALIZABLE`) in the ISO standard, as there is only one possible isolation level, which could be called `SERIALIZED` [21],[22], reduces the number of available actions for integrity constraints by disallowing `NO ACTION` and limiting the extent to which constraints can be `DEFERRED`. The validation step guarantees fully isolated transactions. This means that changes made during a transaction are never visible to other users, but will prevent commit of conflicting transactions.

During a transaction, new records and database objects are temporarily given locations in memory, so that they are accessible and work as expected within the transaction thread. On commit, following the validation step, these objects are relocated in a cascade to the file positions where they will be recorded in the transaction log, and re-installed in the in-memory database. More details of this process are to be found in [23].

The granularity of the test for transaction conflict that is applied in this validation step is that (a) changes to the same database object (other than tables) will always conflict, (b) for tables, we report conflict if any columns read have been updated by another transaction, but if only specific rows have been read, we can limit the validation step to these rows. Validation for this level of granularity is practical even in situations of high concurrency [25]. The most recent implementation of this test (August 2022) uses two simple tree structures for columns and rows for any affected table, and also demonstrates correct behavior for cascades, constraints and triggers (files in [23] have been updated to show this).

For the best implementation of the optimistic concurrency control implied by the existence of the validation step in the commit algorithm, we advocate the use of shareable data structures. When discussing the sharing of modifiable data such as arrays, computer science textbooks often contrast the two approaches of copy on read and copy on write. From our point of view both are wasteful of time and resources, and the use of shareable data structures provides a different approach, which is well suited for the many tree-like structures found in database technology. A good way of motivating the concept is to consider the implementation of strings in programming languages.

In Unix, traditionally, strings (char *) are mutable: anyone with access to the string can modify individual characters in the string. In Java, C# and Python, strings are shareable: the only way to modify an individual character is to create a new string, so if a string is shared between two threads, any change to the string in one thread is not seen in the other thread unless it is explicitly given the new version.

Apart from strings, the most popular data structure in database technology is the B-tree, where each node apart from the root has at least n children and not more than 2n, where n>1, and information is placed in the leaves. In order to make database structures shareable, therefore, the key step is to use a shareable sort of B-tree. The model for this dates from 1982 [26], and the illustration reproduced in Figure 1 below shows that when a change to a tree is made to a leaf, we get a new root and the change requires $O(\log_n N)$ new nodes, where N is the number of leaves.

This means that the old and new version continue to share most of the nodes of the structure. With a little thought we can see that this is more storage-efficient than any of the approaches mentioned above (string implementation, copy on read, copy on write), but imposes a greater load on memory allocation and garbage collection. Crucially though, it is safe, and if we use this kind of structure for to implement all of the indexes and lists in the database many database operations such as starting a new transaction are made much simpler [27]. We return to these aspects in Section VII below.

The DBMS should specify and provide auditing support for a security model that allows local management. There is an opportunity for the SQL standard to encourage good practice in this area. PyrrhoDB has implemented the following practical steps for the local database:

A. Maintenance of the full transaction log as the only artefact placed in non-volatile memory.

There were good reasons for placing volatile information in non-volatile storage in 1972, but they are not valid now. It is understandable that where a database occupies large amounts of physical storage, a database administrator would regard the additional storage required for a transaction log as a luxury. PyrrhoDB's full transaction log is also serialized, so that it is evident that concurrent transactions have been correctly handled. Even in situations of high concurrency, the algorithms and solutions offered here have been shown to be practical [21].

When the only data written to disk is the inserted or updated record, or an indication that a record has been deleted, disk activity required for database traffic is drastically reduced, especially where the database has indexes that are stored on disk [12].

B. Recording the user and role for each change to the database

This is relatively easy to implement, though strongly resisted by database professionals and accountants, who dislike leaving their fingerprints all over the databases they administer or client account they prepare. However, it requires several departures from the SQL standard [2]: its features F771 and F321 allow the "current user" to be declared in the query language rather than being guaranteed by the operating system, and it does not demand that a user sets a single role. For forensic purposes, and to allow staff to substitute in different roles (due to illness etc) it is important to identify both user and declared role and is a simple matter if the transaction log is being maintained as suggested in Section III.A above.

In order to make the role and user information useful for forensic analysis, the grant of object ownership and role usage to roles should be deprecated, and the grant of anything other than these privileges to users should be deprecated.

As suggested in Section III.C, it should be possible to use the definer's role of an object to grant ownership to another user.

C. Database objects should be modified only by their owner, and all execution should use definer's role

From III.B, when objects are defined there is a current role: this is the definer's role, and it must be one of the roles that the user is permitted to use. This role and the owner's identity become properties of the object and can be modified by grant. The details of the new definition are checked both during parsing on every subsequent execution of the object.

The SQL standard specifies a context stack for procedure invocation, so it is again relatively easy to extend the use of such a stack for access to table columns, the sources of views, and the execution of constraints and triggers.

The execution engine then simply sets the current role for the called context to that of the definer of the table, view, procedure, constraint, or trigger, which it knows because of III.B. The invoker still needs appropriate permissions to initiate the process (by accessing or modifying the table or view or calling the procedure) and to access the columns of any table or row result.

The specifications in the standard make it very difficult to create a usable set of permissions for database operations, because users require usage permissions on every data type and column.

Two additional simplifications are recommended: the REFERENCES privilege in the standard then becomes redundant as it becomes the same thing as SELECT, and it simplifies the security model if all data types are usable by PUBLIC (though there may be restrictions on access to their fields if any). Using definer's role as described here, together with these changes, make the security model much easier to operate. New objects can be owned by the user that defines them (with their declared role as the definer's role) and the granting of privileges on an object does not need to consider data types or dependent definitions. Thus, it is much easier to maintain a usable set of privileges on even a large set of database objects.

With these provisions, Pyrrho's security model is simpler to administer and check for validity, but of course it makes execution somewhat slower: to check access permission on a single object requires a single access to the tree of properties of the object, which is typically of depth 3 (see below).

We believe this is an improvement on the arrangements used in Oracle [28] and PostgreSQL [29]. The cautionary words used about definer's role by these products are correct since they are installing native external procedures. Execution by the database server is safe because it can check all object permissions as they are accessed.

By using the role declaration model discussed above, all security settings for a relational database can and should be managed by the database itself, rather than in the database applications. The standard SQL model allows for hierarchical delegation of management of roles and permissions, separate from the authentication of users.

For example, consider the following simple database for a table-tennis club. It allows select access to the two tables shown, but changes to the database by ordinary members must be done with the help of the two procedures provided:

```
create table members (id int primary key, firstname char)
[create table played (id int primary key,
  winner int references members,
  loser int references members, agreed boolean)]
grant select on members to public
grant select on played to public
[create procedure claim(won int, beat int)
  insert into played(winner, loser)
    values(claim.won, claim. beat)]
[create procedure agree(p int)
  update played set agreed=true
   where winner=agree.p and loser in
    (select m.id from members m where user like
       '%'||firstname escape '^')]
create role admin
create role membergames
[grant execute on procedure claim(int, int) to role
```

```
  membergames]
grant execute on procedure agree(int) to role membergames
grant membergames to public
```

To use the given procedures, a member of the public who is allowed to login to the system should set their role to membergames.

IV. The Type System and Metadata

A major difficulty in both enterprise data integration and data collaboration is the definition of a data model that supports application development in different parts of the enterprise. We consider it useful for databases to provide as much support for data semantics where possible, while retaining as much flexibility as possible for local development.

As a first step, we introduce the primitive Document type for JSON values and allow the braces '{' and '}' to delimit row values in SQL, the brackets '[' and ']' as string subscripts for Document values and a built-in Document-valued function HTTP whose parameters are the verb and url, with an optional third parameter being a Document for posted data.

Many DBMS have found the need to embellish their data access methods and database applications in various ways:

- Controlling XML and JSON output for queries, to identify whether table columns are output as attributes/fields or children/subdocuments of the table.
- For data visualization, e.g., charts
- Entity data models: Declaring classes in a database application corresponding to base tables in the database, with derived class references associated with foreign keys, lookup functions etc.

We consider it is good practice to include all such metadata in the database design, and it should be done on a per-role basis, to allow for suites of database applications for different business purposes.

In PyrrhoDB, we have come up with a list of useful metadata identifiers.

```
   Metadata  =  CAPTION | LEGEND  |  X | Y |
((HISTOGRAM | LINE | PIE | POINTS) ['(' id ',' id ')'])
   | ([URL | MIME | SQLAGENT | USER | PASSWORD] string)
| JSON | CSV | ETAG | MILLI
   | MONOTONIC | ((INVERTS|FORMATS) id)
   | ATTRIBUTE | ENTITY | ((SUFFIX|PREFIX) id) | iri .
```

This syntax is a Pyrrho extension, and metadata can be added to a database object (or dropped) by almost any DDL command. Most of the options affect query output for a role in Pyrrho's Web service. The above list provides a rough grouping of these keywords into four groups: (1) data visualization for specific tables and views, (2) provision for collaboration with remote data, (3) provision for adapter functions, and (4) support for local data models. ATTRIBUTE if present for a column indicates a preference for XML output for the containing table. HISTOGRAM, LEGEND, LINE, POINTS, PIE (for table, view or function metadata), CAPTION, X and Y (for column or sub-object metadata) specify JavaScript added to HTML output to draw the data visualizations specified. The syntax allows a string for a description. For INVERTS the id should be the name of the function being inverted, while for FORMATS the id is a type. PREFIX and SUFFIX define ids added to the client output string and in SQL triggers a default constructor for the type, as explained in the currency example at the end of this section.

Pyrrho helps with data visualizations defined using the keywords in group (1) above, using a simple URL-mapped HTTP service, as the following example shows:

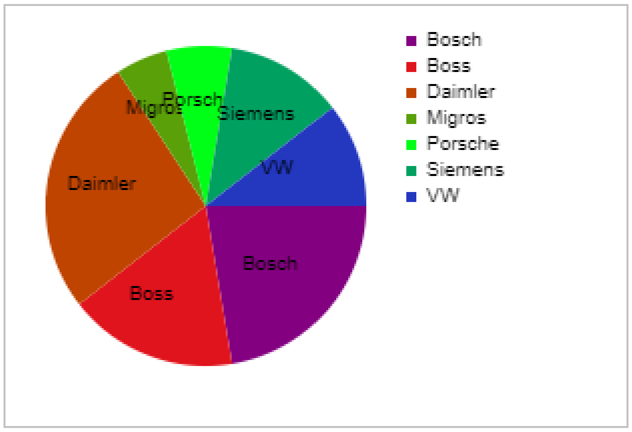
With the database E created by

```
[create table sales (cust char(12) primary key,
custSales numeric(8,2))]
[insert into sales values ('Bosch' , 17000.00),('Boss'
,  13000.00), ('Daimler',20000.00)]
[insert into sales values
('Siemens',9000.00),('Porsche', 5000.00), ('VW',
8000.00), ('Migros' , 4000.00)]
create role E
grant E to "usermachine/username"
```

The data visualization output uses HTML returned to the client application or for immediate display. Here, if the browser is asked for

```
http://localhost:8180/E/E/SALES/?PIE(CUST,CUSTSALES)LEGEND
```

The browser will display the following output from the PyrrhoDB server:



We return to this example below.

User-defined types can nominate a primitive type in the UNDER clause, and this can be useful for distinguishing data that has been imported or used in different suborganisations. The SQL standard already provides the OF predicate for selecting a value of a type, a TREAT function for specifying the subtype for a scalar value, and a "create table of type" mechanism for specifying row types. Pyrrho adds the ability to specify a subtype for VALUES.

As an example of the resulting syntax, if we defined:

```
[create type currency as(amt numeric,unit char)
  method exchange(tounit char) returns currency,
  method tonumeric() returns numeric]
```

The exchange method here would be implemented for the database using the above-mentioned HTTP function. There are many currency converters available on the Internet, for example

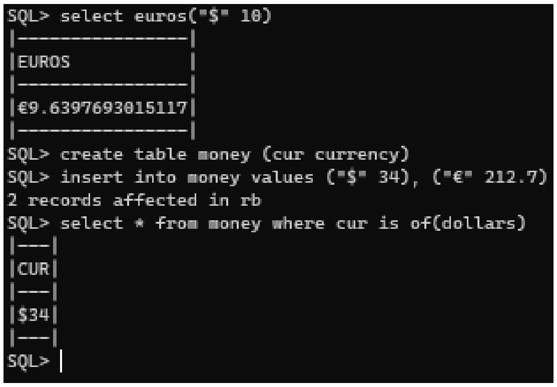
```
[create method exchange(tounit char) returns currency
for currency
    begin
      if unit=tounit return this;
      declare rates document;
      declare roe numeric;
      set rates=http('post',
'http://www.floatrates.com/daily/'||unit||'.json');
      set roe=rates[lower(tounit)]['rate'];
      return currency(amt*roe,tounit)
    end]
```

Then we could have

```
[create type dollars under currency check(unit='USD')
   constructor method dollars(x numeric),
   constructor method (x currency) prefix "$"]
[create constructor method dollars(x numeric)
   begin set amt = x; set unit = 'USD' end]
[create constructor method dollars(x currency)
   begin set amt=x.exchange('USD').amt;
   set unit='USD' end]
```

If we have similar declarations for euros, we could write things as simple as

```
    select euros("$" 10)
    create table money (cur currency)
    insert into money values ("$" 34), ("€" 212.7)
    select * from money where cur is of(dollars)
```



```
SQL> select euros("$" 10)
|----------------|
|EUROS           |
|----------------|
|€9.6397693015117|
|----------------|
SQL> create table money (cur currency)
SQL> insert into money values ("$" 34), ("€" 212.7)
2 records affected in rb
SQL> select * from money where cur is of(dollars)
|---|
|CUR|
|---|
|$34|
|---|
SQL> |
```

We give an example using the data model metadata directive ENTITY in Section VIII below.

V. VIEW-MEDIATED REMOTE ACCESS

Data warehousing involves creating central data repositories (using extract-transform-load technologies) to enable analytic processing of a combined data set. There are several situations where this is undesirable, for example where the resulting data protection responsibility at the central repository is excessive, where the data is volatile and it becomes expensive to maintain all of the centrally-held data in real time, or where it is better to leave the data at its sources where the responsibility lies [10]. With database technology, a View (if defined but not materialised) allows access to data defined in other places. The virtual data warehouse concept exploits this notion, and endeavours to avoid the central accumulation of data. Pyrrho uses HTTP to collect data from the remote DBMS using a simple REST interface [22], and so the resulting technology here is called RESTView.

Thus, with RESTView, a Pyrrho database allows definition of views where the data is held on remote DBMS(s), and is accessible via SQL statements sent over HTTP with Json responses. Pyrrho itself provides such an HTTP service and the distribution includes suitable interface servers (RestIf) to provide such a service for remote MySQL and SqlServer DBMS. The implementation allows for authentication as an ordinary client of the remote DBMS, whose administrator can grant access to a suitably defined view.

The HTTP access provides the user/password combinations set up for this purpose within MySQL by the owners of contributor databases. In the use cases considered here, where a query Q references a RESTView V, we assume that (a) materializing V by Extract-transform-load is undesirable for some legal reason or because of the high data volumes required, and (b) we know nothing of the internal details of contributor databases. A single remote select statement defines each RESTView: the agreement with a contributor does not provide any complex protocols, so that for any given Q, we want at most one query to any contributor, compatible with the permissions granted to us by the contributor, namely grant select on the RESTView columns.

Crucially, though, for any given Q, we want to minimize the volume D of data transferred. We can consider how much data Q needs to compute its results, and we rewrite the query to keep D as low as possible. Obviously, many such queries (such as the obvious select * from V) would need all of the data. At the other extreme, if Q only refers to local data (no RESTViews) D is always zero, so that all of this analysis is specific to the RESTView technology.

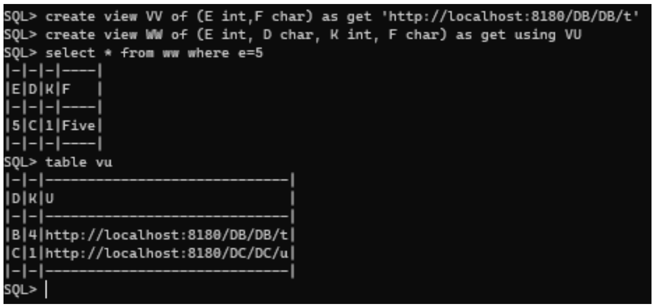
During query processing views are replaced by their definitions, so that the overall query becomes a selection from the tables they reference. The process deals with the situation that a table can be referenced in more than one place by adding unique identifiers for each table reference.

Filters are applied at the lowest level of the query (e.g., directly on a remote table), and traversal of a remote table creates a roundtrip of the REST service to the given URL. The JSON representation of the result returned is slightly enhanced to add the registers used to compute any remote aggregations [23].

The syntax is

ViewDefinition = [ViewSpec] AS
    (QueryExpression | GET [USING Table_id]) {Metadata}.

The alternative shown by the vertical bar corresponds to whether the view has one single contributor or multiple remote databases. The QueryExpression option here is the normal syntax for defining a view. The REST options both contain the GET keyword. The simplest kind of RESTView is defined as GET from a url defined in the Metadata. The types of the columns need to be specified in a slightly extended ViewSpec syntax. If there are multiple remote databases, the GET USING table_id option is available. The rows of this table describe the remote contributions: the last column supplies the metadata for the contributor including a url, and data in the other columns (if any) is simply copied into the view. For example:



```
SQL> create view VV of (E int,F char) as get 'http://localhost:8180/DB/DB/t'
SQL> create view WW of (E int, D char, K int, F char) as get using VU
SQL> select * from ww where e=5
|-|-|-|----|
|E|D|K|F   |
|-|-|-|----|
|5|C|1|Five|
|-|-|-|----|
SQL> table vu
|-|-|-----------------------------|
|D|K|U                            |
|-|-|-----------------------------|
|B|4|http://localhost:8180/DB/DB/t|
|C|1|http://localhost:8180/DC/DC/u|
|-|-|-----------------------------|
SQL> |
```

Depending on how the remote contributions are defined, RESTViews may be updatable, and may support insert and delete operations.

The implementation of these ideas was demonstrated in [23].

With these arrangements it is important to consider transaction requirements for multiple-host scenarios. The fundamental difficulty is the so-called two-army problem, according to which all data needs a single transaction master. Every transaction is initiated at one database (call its server's host local), and then accesses remote data via a view definition of the type described above. The transaction can commit changes on the local server and at most one remote server update, assuming the transaction provides suitable credentials for that database. The commit takes place according to the following mechanism (a) the local database is locked, (b) the local changes are validated, (c) HTTP 1.1 is used to perform the single remote update (using the RFC7232 mechanisms), (d) then the local commit can complete and unlock. With just one remote update this mechanism is safe and can be rolled back on any exception.

It is possible to imagine interworking between heterogeneous DBMS using these techniques, so that it is important to maintain the use of standard industry approaches for REST services. Many systems have implemented a URL and XML/JSON to database mapping, and the ETag mechanism from RFC7232 [24] can be leveraged to provide transactional features [20]. Currently in Pyrrho there are several options for this, determined by the metadata flags `URL` and `ETAG` listed above.

Consider again the sales database E from Section IV, which over time gains a great many sales records. Suppose E offers to role rs_V a view into the data that includes a computation of the current runningSalesShare as a number between 0 and 1:

```
[create view sales_V(cust, custSales, runningSalesShare)
 as select cust, custSales,
  (select sum(custSales) from sales where custSales >=
u.custSales) /
   (select sum(custSales) from sales)
from sales as u]
create role rs_V
grant rs_V to "user\machine"
```

Then this view can be accessed from the named machine using dashboard-style queries that categorize the customers A, B or C depending on the current runningSalesShare without having to be told all of the individual sales.

```
[select case when runningSalesShare <= 0.5 then 'A'
  when runningSalesShare > 0.5  and
   runningSalesShare <= 0.85 then 'B'
  when runningSalesShare > 0.85 then 'C'
  else null
  end as Category,
 cust, custSales,
 cast(cast(custSales  /  (select  sum(custSales)  from
sales_V) * 100
   as decimal(6, 2))
  as char(6)) || ' %' as share
from sales_V
order by custSales desc]
```

The output, and a pie chart derived from it, are shown in Figure 2.

VI. IMPLEMENTATION USING SHAREABLE DATA STRUCTURES

This section provides some details of the implementation for the above features, following the philosophy outlined above using globalized and architecture-independent data formats. PyrrhoDB uses 64-bit uids for all database objects, log entries, and table rows. It uses a representation for variable length primitives Integer (up to 2040 bits), Real (Integer mantissa, int scale) and Char (Unicode strings up to 2^{60} bytes). The naming of database objects (except data types) is on a per-role basis.

As mentioned above, the database is represented on disk by a transaction log, consisting of a sequence of "physicals": there are roughly 70 physical formats: one of these is for transaction details, another for a table identifier, another for column details etc. The details are in the Pyrrho manual in the Github distribution. The transaction log uses append storage.

On first access by the server a database's entire transaction log is read and the live database objects constructed in memory.

Following the success of StrongDBMS [21] in performing serializable transactions in a high-concurrency demonstration, PyrrhoDB has been re-implemented to use shareable data structures throughout. A shareable data structure cannot be updated or modified, so any change involves creation of a new instance. Examples of shareable data structures are primitive data types such as integer or float, the string type in C#, Java or Python, and classes whose fields are all readonly shareable data structures. With the help of some simple shareable building blocks (BList<V> and BTree<K,V>) it is straightforward to build up shareable data structures representing tables, indexes and even databases.

A Domain class specifies a base type and many other properties. If it has columns (e.g., a user-defined type or base table) the Domain will also specify a list of column uids, and a tree giving the Domain of each field. The primitive types have system-allocated (negative) uids, and any other Domain is created as physical objects in the database that defines it.

Table rows are composed of TypedValues: a TypedValue is defined by a Domain and a shareable data structure.

The BTree<K,V> implementation was described above: it is an unbalanced B-Tree that gives worst-case O(logN) performance for inserting, changing or deleting a node. Any of these changes creates a new root node and new internal nodes to the new leaf node, making at most logN new nodes, while the rest of the nodes are shared between the old and new version of the tree. This is therefore surprisingly efficient.

BList<V> is not so clever. It is implemented as a BTree, but it renumbers its nodes 0, 1, 2, … resulting in a worst-case performance of O(N).

BList and BTree are shareable data structures provided their contents (all K and V objects) are shareable. Instead of the enumerators found in Java and C#, traversal of BTrees and BLists uses “bookmarks” that are also shareable data structures: two-way traversal is possible, and traversal continues to traverse from the root it was given and so is unaffected by changes to the tree it is traversing.

All classes that make up database objects are shareable. For example, Rowsets are basically a BTree of rows that are TypedValues, traversed by Cursors, which are a subclass of the bookmark class mentioned above.

A new server thread is started for each connection to a database. Protocol requests typically create a Transaction to query or modify the database or read the next group of data from the result of a query, which is confined to the connection thread The creation of a transaction is a simple matter: each transaction starts with a copy of the root node of the database (a snapshot). On rollback or disconnect, the transaction can simply be forgotten, as no other thread has seen it.

VII. Query processing and Compiled objects

During parsing, uids are allocated for the resulting expressions, and for anything that may be committed as a new database object. Uids in the range above (currently) 4×2^{60} are allocated as required: there are several ranges for these depending (for example) on whether their lifetime is the current session, the current transaction, or the current lexical input. SQL expressions all have Domains discovered during parsing, and RowSets all have Domains that specify their columns, so that ad-hoc Domains are constructed as required during query processing. Since a query may reference a table source more than once (via a TableRowSet), the column uids for TableRowSets need to be specific to such a reference and are allocated in the heap range (above 7×2^{60}): this process is called instancing in the implementation. Views may also reference more than one source, so the instancing process also applies to them. A similar requirement exists for table-valued functions.

In this section we also consider how the concept of local data management can be realized. As mentioned in Section III.C, the server should remain in control of execution of stored procedures, triggers, and constraints, so that such features should be written in SQL. Since the definer of a compiled object generally has different privileges from the user making a query or update, it is important to ensure that executable code is compiled in advance. For reasons of forward and backward compatibility the database file contains only the SQL source code for stored procedures, constraints, triggers, etc. The compiled components are constructed when the database is loaded in the server (after a cold start).

As mentioned above, many database objects correspond to permanent physical records in the transaction log, and so their defining position is fixed and they can be shared with all transactions for this database. Objects constructed by the server during compilation (also in fact shareable) do not have physical file positions, so instead receive uids in a dedicated range (currently 6×2^{60} .. 7×2^{60}-1), and form a collection stored with the in-memory version of the compiled object. Most compiled objects contain executable code, but this mechanism is also used for the Domain of a base table or view. The actual uids allocated to these compiled objects will depend on the order of the physical objects in the log, and will depend on the current version of the server. During instancing, column uids will be allocated in a cascade, since many compiled objects will contain references to the columns being instanced.

Cascades are also used in the process of RowSet review, in which the RowSet pipeline is simplified wherever possible based on the existence of indexes and filters that were not available at compilation time.

VIII. The Versioned library and data models

The above discussion described how the data model for a database could be represented in the database implementation. The real benefit of placing the data model in the database is to make it available to the application programmer, so that all applications targeting a database can agree on the structure and semantics of its data. At present, Pyrrho provides such support for applications written in C#, Java, and Python, in addition to a thread-safe version of the Command/ExecuteReader/Read programming interface familiar from ADO.NET and JDBC.

The current implementation was inspired by Microsoft’s Entity Framework [30] and Java Persistence Architecture [31] but differs from these in the crucial proviso that application programmers should start with class definitions generated by (and at runtime checked by) the server, rather than writing their own version of the model in the form of annotations or code attributes.

The following example illustrates the type of support available. Suppose a database ABC contains a role “Sales” that defines the following tables:

```
[create table "Customer"(id int primary key, "NAME"
      char unique)]
[create table "Order"(id int primary key, cust int
      references "Customer", "OrderDate" date, "Total"
      numeric(6,2))]
```

Then the system table "Role$ClassValue" will provide code fragments similar to the following:

```
using System;
using Pyrrho;

/// <summary>
/// Class Customer from Database ABC, Role Sales
// PrimaryKey(ID)
// Unique(NAME)
```

```
/// </summary>
[Table(23,122)]
public class Customer : Versioned {
[Field(PyrrhoDbType.Integer)]
[AutoKey]
  public Int64? ID;
[Field(PyrrhoDbType.String)]
  public String? NAME;
  public Order[] orders =>
     conn.FindWith<Order>(("CUST",ID));
}
/// <summary>
/// Class Order from Database ABC, Role Sales
// PrimaryKey(ID)
// ForeignKey, RestrictUpdate, CascadeDelete(CUST)
/// </summary>
[Table(175,362)]
public class Order : Versioned {
[Field(PyrrhoDbType.Integer)]
[AutoKey]
  public Int64? ID;
[Field(PyrrhoDbType.Integer)]
  public Int64? CUST;
[Field(PyrrhoDbType.Date)]
  public Date? OrderDate;
[Field(PyrrhoDbType.Decimal,"Domain    NUMERIC    Prec=6
Scale=2")]
  public Decimal? Total;
  public Customer customer =>
     conn.FindOne<Customer>((“ID”,CUST));
}
```

The numbers 23 and 175 are references to the defining positions of these objects in the database, and the other numbers are schema keys, which will be checked by the server when the application runs to ensure that the table definition has not changed. We can see that the columns defined for the table are publicly accessible in these classes (while the server will check the user and role on access), and Pyrrho's data types of these columns are provided as attributes.

Importantly, the foreign key relationship between the tables has resulted in two additional "navigation" fields in the classes above, providing quick access to the customer for an order, and the orders for a customer. The primary and unique key declarations also allow quick access.

A simple program to use the above class definitions could begin

```
static void Main
{
 conn = new PyrrhoConnect("Files=Demo;Role=Sales");
 conn.Open();
 try
 {
// Get a list of all orders showing the customer name
   var aa = conn.FindAll<Order>();
   foreach (var a in aa)
    Console.WriteLine(a.ID + ": " + a.customer.NAME);
   if (aa.Length == 0)
   {
    Console.WriteLine("The Order table is empty");
    goto skip;
   }
 // change the customer name of the first
 // (update to a navigation property)
   var j = aa[0].customer;
   j.NAME = "Johnny";
   j.Put();
 // add a new customer (autokey is used here)
   var g = new Customer() { NAME = "Greta" };
   conn.Post(g);
 // place a new order for Mary
 // (secondary index, single quotes optional here!)
   var m = conn.FindOne<Customer>(("NAME","Mary"));
   var o = new Order()
    { CUST = (long)m.ID,
      OrderDate = new Date(DateTime.Now) };
    conn.Post(o);
```

The Versioned base class above uses ETags, allowing the library to associate object references in the code to rows in the database, and this enables the shorthand notation in the above sample program, in addition to providing automatic transaction validation when committing an explicit transaction is started, using an API similar to ADO.NET and JDBC. For further details see the Pyrrho manual [32].

IX. Conclusions

This paper has reviewed a number of desirable changes to the relational database model that have been signaled in recent literature and outlined implementations of these improvements that can be found in the ShareableDataStructures project on Github [32]. The implementation of PyrrhoDB v7 is currently at the alpha stage and feedback on these ideas is welcomed. The authors are grateful for the many expressions of support and encouragement we have received during this project.

References

[1] Crowe, M. K. and Laux, F.: Data Evolution and Durabiility: Lessons from the PyrrhoDB experiment, keynote speech, ICSEA 2022, Lisbon, Portugal October 2022, https://www.iaria.org/conferences2022/filesICSEA22/Keynote_MalcolmCrowe_FritzLaux_DataEvolutionAndDurability.pdf (accessed 4 December 2022).

[2] ISO. (2016). IEC 9075: 2016: Information technology: Database languages: SQL, International Organization for Standardization.

[3] Schüle, M. E., Kemper, A., and Neumann, T. (2022, July). Recursive SQL for Data Mining. In 34th International Conference on Scientific and Statistical Database Management (pp. 1-4).

[4] Stockinger, K., Bundi, N., Heitz, J., and Breymann, W. (2019). Scalable architecture for Big Data financial analytics: user-defined functions vs. SQL. *Journal of Big Data*, *6*(1), 1-24.

[5] Khine, P. P. and Wang, Z. (2019). A review of polyglot persistence in the big data world. *Information*, *10*(4), 141.

[6] Aluko, V. and Sakr, S. (2019). Big SQL systems: an experimental evaluation. *Cluster Computing*, *22*(4), 1347-1377.

[7] Antonopoulos, P., Budovski, A.,Diaconu, C., Hernandez Saenz, A., Hu, J… (2019, June). Socrates: The new sql server in the cloud. In *Proceedings of the 2019 International Conference on Management of Data* (pp. 1743-1756).

[8] Crowe, M. K. (2015). The Pyrrho Book, University of the West of Scotland, ISBN 978-1-903978-50-4.

[9] Alam, F. and Kamal, N.(2019): Survey on Data Warehouse from Traditional to Realtime and Society Impact of Real Time Data, Inl Jnl Computer Applications 177.9 p.20-24.

[10] Interlandi, M., Ekmekji, A., Shah, K., Gulzar, M. A., Tetali, S. D. et al. (2018). Adding data provenance support to Apache Spark. *The VLDB Journal*, *27*(5), 595-615.

[11] Crowe, M. K., Begg, C. E., and Laux, F. (2017), Data validation for big live data, In DBKDA 2017, The Ninth International Conference of Advances in Databases, Knowledge, and Data Applications, Barcelona, ISBN 978-1-61208-558-6 (pp 30-36)

[12] Crowe, M. K. (2005): Transactions in the Pyrrho database engine, in Hamza, M. H. (ed.): DBA 2005, Proceedings of the IASTED International Conference on Databases and Applications, Innsbruck, ISBN: 0-88986-460-8 (pp 71-76)

[13] de Moura, M. C., Davalos, V., Planas-Serra, L., Alvarez-Errico, D., Arribas, et al. (2021). Epigenome-wide association study of COVID-19 severity with respiratory failure. EBioMedicine, 66, 103339.

[14] Parsi, M. and Akbarpour Jannat, M. R. (2021). Tsunami warning system using of IoT. Journal of Oceanography, 11(44), 1-17.

[15] Soriano, A., Carmeli, Y., Omrani, A. S., Moore, L. S., Tawadrous, M., and Irani, P. (2021). Ceftazidime-avibactam for the treatment of serious Gram-negative infections with limited treatment options: a systematic literature review. Infectious Diseases and Therapy, 10(4), 1989-2034.

[16] Gorobets, A. and Bakhvalov, P. (2022). Heterogeneous CPU+ GPU parallelization for high-accuracy scale-resolving simulations of compressible turbulent flows on hybrid supercomputers. Computer Physics Communications, 271, 108231.

[17] Bai, Y., Ma, Y., Yang, Q., Florez-Lopez, J., Li, X., and Biondini, F. (2021). Earthquake-induced damage updating for remaining-life assessment of steel frame substructure systems. Mechanical Systems and Signal Processing, 159, 107782.

[18] Komenda, M., Černý, V., Šnajdárek, P., Karolyi, M., Hejný, M., Panoška, P., et al. (2022). Control Centre for Intensive Care as a Tool for Effective Coordination, Real-Time Monitoring, and Strategic Planning During the COVID-19 Pandemic. Journal of medical Internet research, 24(2), e33149.

[19] Meana Llorián, D., González García, C., Pelayo García-Bustelo, B. C., and Cueva Lovelle, J. M. (2021). BILROST: Handling actuators of the internet of things through tweets on twitter using a domain-specific language. International Journal of Interactive Multimedia and Artificial Intelligence.

[20] Finlayson, R and Cheriton, D. "Log files: An extended file service exploiting write-once storage." ACM SIGOPS Operating Systems Review 21.5 (1987): 139-148.

[21] Crowe, M. K. and Fyffe, C. (2019): Benchmarking StrongDBMS, (Keynote speech) at DBKDA 2019, The Eleventh International Conference of Advances in Databases, Knowledge, and Data Applications https://www.iaria.org/conferences2019/filesDBKDA19/MalcolmCrowe_CallumFyffee_Keynote_BenchmarkingStrongDBMS.pdf (accessed 4 December 2022)

[22] Crowe, M. K. and Laux, F.: Implementing True Serializable Transactions, Tutorial video, DBKDA 2021, The Thirteenth International Conference on Advances in Databases, Knowledge, and Data Applications, Valencia, Spain. https://www.youtube.com/watch?v=t4h-zPBPtSw&t=39s, (accessed 4 December 2022)

[23] Crowe, M. K. and Laux, F.: Implementing True Serializable Transactions, Tutorial files, DBKDA 2021, The Thirteenth International Conference on Advances in Databases, Knowledge, and Data Applications, Valencia, Spain. https://www.iaria.org/conferences2021/filesDBKDA21/

[24] Fielding, R. T. and Reschke, J (eds) (2014): RFC 7232: Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests, IETF.org

[25] Crowe, M. K. and Laux, F.: Reconsidering Optimistic Algorithms for Relational DBMS, DBKDA 2020 The Twelfth International Conference on Advances in Databases, Knowledge, and Data Applications, Lisbon, Portugal

[26] Krijnen, T. and Meertens, G. L. T.: "Making B-Trees work for B". Amsterdam : Stichting Mathematisch Centrum, 1982, Technical Report IW 219/83

[27] Crowe, M. K. and Matalonga, S. (2019): StrongDBMS: built from immutable components, In DBKDA 2019, The Eleventh International Conference of Advances in Databases, Knowledge, and Data Applications, Athens, ISBN 978-1-61208-715-3 (pp. 11-16)

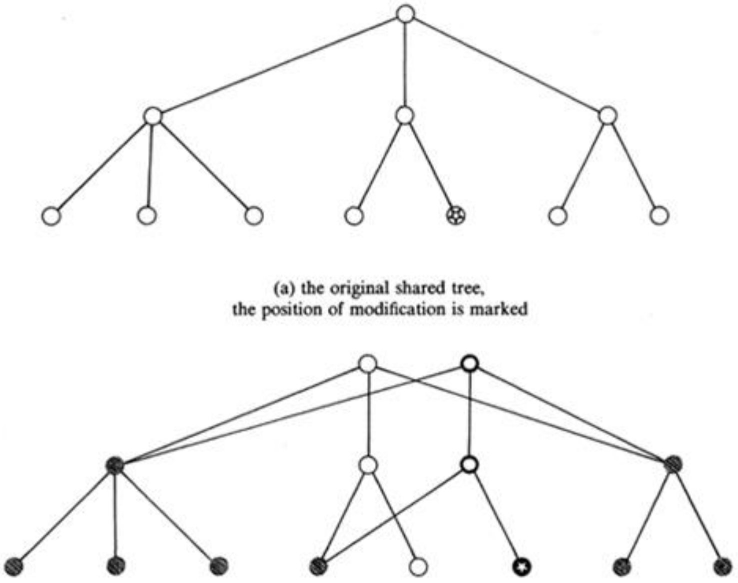
[28] Oracle.com Product Documentation: https://docs.oracle.com/en/database/oracle/oracle-database/19/dbseg/managing-security-for-definers-rights-and-invokers-rights.html (Accessed 4 December 2022)

[29] PostgreSQL Product Documentation https://www.postgresql.org/docs/current/sql-createfunction.html (Accessed 4 December 2022)

[30] Microsoft Product Software on Github: https://github.com/dotnet/efcore (Accessed 4 December 2022)

[31] Oracle Product Documentation https://docs.oracle.com/javaee/7/tutorial/persistence-intro.htm#BNBPZ (Accessed 4 December 2022)

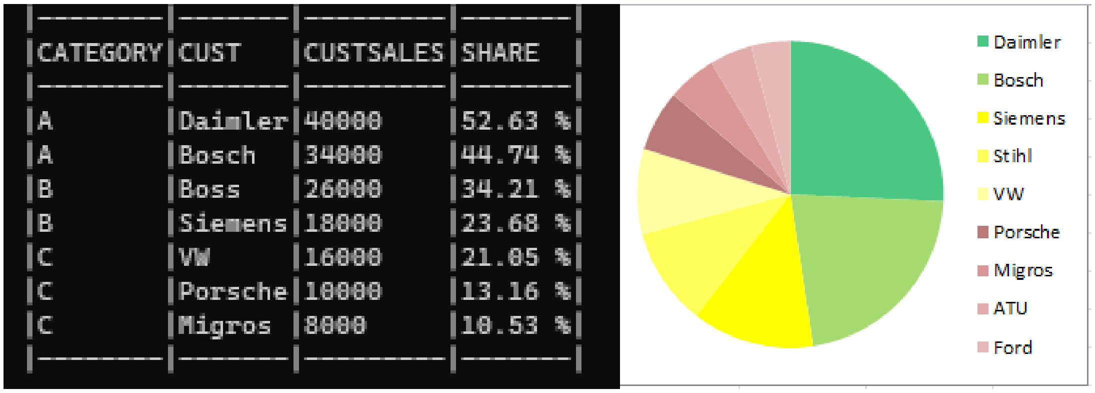
[32] Crowe, M. K.: PyrrhoDB manual on Github; https://github.com/MalcolmCrowe/ShareableDataStructures/blob/master/PyrrhoV7alpha/doc/Pyrrho.pdf (Accessed 4 December 2022)



(a) the original shared tree,
the position of modification is marked

(b) the path to the position of modification is "deshared",
new nodes are thicker, shared nodes are shaded

Figure 1. Operation of B-Trees [26]



```
|--------|-------|---------|-------|
|CATEGORY|CUST   |CUSTSALES|SHARE  |
|--------|-------|---------|-------|
|A       |Daimler|40000    |52.63 %|
|A       |Bosch  |34000    |44.74 %|
|B       |Boss   |26000    |34.21 %|
|B       |Siemens|18000    |23.68 %|
|C       |VW     |16000    |21.05 %|
|C       |Porsche|10000    |13.16 %|
|C       |Migros |8000     |10.53 %|
|--------|-------|---------|-------|
```

Figure 2: ABC analysis from Section VI example (as output from PyrrhoDB client and server)

Designing Context-aware Data Plausibility Automation Using Machine Learning

1st Mohaddeseh Basiri
KTH Royal Institute of Technology
Stockholm, Sweden
mbasiri@kth.se

2nd Johannes Himmelbauer
Software Competence Center Hagenberg GmbH
Hagenberg, Austria
johannes.himmelbauer@scch.at

3rd Lisa Ehrlinger
Software Competence Center Hagenberg GmbH
Hagenberg, Austria
lisa.ehrlinger@scch.at

4th Mihhail Matskin
KTH Royal Institute of Technology
Stockholm, Sweden
misha@kth.se

***Abstract*—In the last two decades, computing and storage technologies have experienced enormous advances. Leveraging these recent advances, Artificial Intelligence (AI) is making the leap from traditional classification use cases to automation of complex systems through advanced machine learning and reasoning algorithms. While the literature on AI algorithms and applications of these algorithms in automation is mature, there is a lack of research on trustworthy AI, i.e., how different industries can trust the developed AI modules. AI algorithms are data-driven, i.e., they learn based on the received data, and also act based on the received status data. Then, an initial step in addressing trustworthy AI is investigating the plausibility of the data that is fed to the system. In this work, we study the state-of-the-art data plausibility check approaches. Then, we propose a novel approach that leverages machine learning for an automated data plausibility check. This novel approach is context-aware, i.e., it leverages potential contextual data related to the dataset under investigation for a plausibility check. We investigate three machine learning solutions that leverage auto-correlation in each feature of dataset, correlation between features, and hidden statistics of each feature for generating the checkpoints. Performance evaluation results indicated the outstanding performance of the proposed scheme in the detection of noisy data in order to do the data plausibility check.**

***Index Terms*—Artificial intelligence; Machine learning; Automation; Plausibility check; Anomaly detection; Ontology; Context-aware.**

I. Introduction

Due to the rapid development of information technology and manufacturing process, traditional manufacturing enterprises have been transformed to the digital and smart factories [1], [2]. This improvement leads to the emerging complex systems with thousands of components and sub-systems, in which continuous monitoring of these systems is of crucial importance. From the data analytic point of view, this means surveillance of large amounts of time series data in order to ensure the correctness of the data and run data plausibility checks. So, regarding the huge amounts of data, human monitoring of data is not feasible, which conducts us to the automated plausibility check using Machine Learning (ML) and data mining approaches [3].

Data plausibility describes the state when data seems reasonable. Conversely, an anomaly or outlier is a data point that is remarkably different from the remaining data. A possible approach for implementing outlier detection is to run plausibility checks [4]. Rapid and efficient outlier detection is critical for many applications including intrusion detection systems, credit card fraud, sensor events, medical recognition, law enforcement, etc. [5]. Although outlier detection is an intensively researched topic in the machine learning and statistics community [6], there are still many open challenges in practice. The first challenge is context dependence. For example, a very high fluctuation rate in a company dataset might be reasonable for a catering service, but not for a construction company. Thus, the decision of whether a data sample seems reasonable (i.e., it is not an outlier) often depends on the context within it appears. Second, the high dimensionality of the dataset creates difficulties for data plausibility check [7]. Since the number of features increases in a high-dimensional dataset, the amount of data for accurate generalization also raises, which results in data sparsity and scattering. This data sparsity is because of inessential features or irrelevant attributes that hide the correct anomalies. So, anomaly detection is becoming a challenging task by increasing the number of features and attributes in large datasets. In addition to these challenges, there are some inherent issues such as difficulties in the design of threshold between normal and anomalous data, and much noise existence due to incorrect measurements or sensor malfunctioning that may cause the false notifications. On the other hand, data imbalance as the common problem in anomaly detection approaches affects the robustness of models, as very few outlier samples are available.

In order to address the aforementioned challenges, we present a novel context-aware approach for an automated data plausibility check, where there is a lack of research in the literature. In this approach, machine learning techniques are leveraged on top of semantic models, e.g., ontology, and benefited from side information in the datasets. Semantic data models like ontology [8] facilitate the incorporation of semantic information into the data. This work is the extended version of [1]. The focus of this paper is on multivariate outlier detection on the level of records (i.e., samples, rows) instead of single values. In this regard, the main contributions of this work include:

1) Presenting a data plausibility check framework; including test ontology, test data generator, checkpoint, and their

message exchanges.

2) Disclosing three types of tests, to be deployed in the test ontology, executed in the test generator, and used in decision making in the checkpoint module. These tests include:
   a) Inter-feature check, checking features based on their relations leveraging an machine learning module for prediction of a feature from some related features (list of neighbors is given by the test ontology from training)
   b) Intra-feature check (1), checking a feature based on its lags (previous values) using an ML module for prediction based on the lags (number of lags is given by the test ontology from training),
   c) Intra-feature check (2), checking a feature leveraging metadata and its long-term statistics (the type of needed metadata and action on them are given by the test ontology)
3) Presenting a comprehensive analysis of the performance of the proposed solution on a propriety dataset and drawing insights and conclusions from the analyses.

The rest of this paper is organized as follows: Section II presents state-of-the-art anomaly detection techniques. Section III describes the needed background for the work in more details. Section IV presents the data and models used to solve the problem. Section V describes our solution for solving the problem. Simulation results and discussion are presented in Section VI. In Section VII, the findings of this work are presented in a brief but succinct manner.

II. Related Work

Anomaly detection, as the concept of identifying patterns or data points that are significantly different from the expected behavior, has been widely studied. State-of-the-art using anomaly detection algorithms can be categorized as following [9]:

Classification Based: This algorithm strives to discern normal data instances from the abnormal ones in the given dataset space by using a trained model. It is categorized into one-class and multi-class models. In one-class models, a distinguished threshold is learned to label data points outside of this threshold as anomalies instances [10]. In multi-class models, multiple classifiers are trained. A data point is recognized as an anomaly if none of the classifiers can label it as the normal instance [11]. Various anomaly detection techniques such as neural networks, Bayesian networks, support vector machines, and rule-based utilize different classification algorithms to build their classifiers.

Nearest Neighbor Based: In this technique, normal data points are in compact neighborhoods, while anomalous data points are far from their nearest neighbors. This technique needs a distance or similarity measurement between two data points in order to recognize which data points are far from or different from other points. For continuous features, Euclidean distance is used, and for categorical features, a simple matching coefficient is a common option. In multivariate data points, the combination of computed distance for each feature is usually leveraged. The nearest neighbor technique is categorized into two groups regarding how they compute the anomaly score: 1) The distance of a data point to its k^{th} nearest neighbor is used as the anomaly score, e.g., k-nearest neighbor approach [12]. 2) The relative density of each data point is computed as the anomaly score, e.g., Local Outlier Factor (LOF) [13].

Clustering Based: In this algorithm, similar data instances are grouped into clusters. There are three categories of clustering-based anomaly detection techniques. First, techniques that suppose normal data instances belong to a cluster, while abnormal data points do not belong to any cluster, e.g., SNN clustering [14]. Second, algorithms that consider normal data instances are near to the closest cluster centroid, while outliers are far from their closest cluster centroid, e.g., Self-Organizing Maps [15]. Third, those assume normal data instances create large and dense clusters, while anomalous data points create small or scattered clusters, e.g., Cluster-Based Local Outlier Factor (CBLOF) [16].

Statistical: Regarding the basic assumption of statistical anomaly detection techniques, a data point is anomaly if it is not generated by the stochastic model. In other words, normal data points happen in high probability areas of a stochastic model, while outliers happen in the low probability areas of the stochastic model. In these approaches, a statistical model (usually for normal patterns) is applied to the dataset and then a statistical inference test is utilized to identify whether a data point fits well to this model or not. Regarding the applied test statistic, data instances that there are low probability to be created from the learn model are considered as anomalous data. Parametric and non-parametric techniques are two approaches that can be leveraged to fit a statistical model. Parametric techniques benefit the distribution knowledge and compute parameters from the given data, while non-parametric techniques do not. Gaussian model based algorithms like Maximum Likelihood Estimation (MLE) [17], regression model based like Auto-regressive Integrated Moving Average (ARIMA) [18], and combination of parametric distribution based algorithms like Expectation Maximization (EM) [19] are instances of parametric techniques. Histogram based such as Intrusion-Detection Expert System (IDES) [20], and kernel function based like parzen windows estimation [21] are samples of non-parametric techniques.

Information Theoretic: In this approach, the information content of the dataset is analyzed. The purpose of this technique is to solve a double optimization problem in order to determine the minimized subset that maximizes the complexity reduction of the dataset, and finally label that subset as the outlier. Entropy and Kolmogorov Complexity [22] are two examples of this category.

Spectral: This technique tries to find a lower-dimensional subspace in such a way that outliers and normal data points are remarkably different. Hence, anomalies can be easily distinguished. Principal Component Analysis (PCA) is used in many techniques in order to project data points into a lower dimensional space [23].

In order to have better overview of different techniques and their algorithms, advantages and disadvantages of each techniques are summarized in Figure 1.

III. Background

In this section, ARIMA, decision tree, and random forest as machine learning algorithms and ontology are described in more details.

A. ARIMA

ARIMA (Auto-Regressive Integrated Moving Average) is an extension of an auto-regressive moving average (ARMA) model. Both of these models are utilized in order to have better understanding of time series data or predict future values of an attribute [18]. The AR part of ARIMA shows that the attribute of interest is regressed on its own lagged (i.e., on its prior values). The MA part is representation of the regression error, which is the linear combination of contemporaneous error values and errors at various times in the past. The I (for ”integrated”) shows that the data values have been substituted with the discrepancy between their values and the previous values. ARIMA model is denoted by $ARIMA(P, I, Q)$, where P is the order of auto-regressive model (number of time lags), I is the degree of differencing, and Q is the order of moving-average model. The aim of each of these features is to make the model fit well with the data.

B. Decision Tree and Random Forest

Decision tree as a rule-based classifier corresponds each internal node of the tree to an attribute. Each branch of the tree represents a condition (rule) on the related attribute. The result of the condition on the related attribute can be binary, categorical, or real-valued. Depending on the result of the condition, a test example pursues the related branches starting from the root node and moves down to a leaf node. Leaf nodes represent the labels, which are the results of classification. The basic idea of a single decision tree is leveraged for random forests (RF)s and ensemble learning. Regarding the main principle, utilizing an ensemble of several naive weak classifiers can cause to a much more powerful classifier, such that each of this unique weak classifier can perform rather more powerful than random estimation and independent of all other classifiers [24].

As shown in Figure 2, random forest works based on the bagging algorithm and uses ensemble learning technique. It builds as several trees as possible on the subset of data and merges the results of all the trees together. In this way, it decreases overfitting problem and also reduces the variance and hence improves the accuracy. This classifier can handle missing values and does not need feature scaling. Random forest is usually stable to outliers. Even if a new data instance is inserted in the dataset, the entire algorithm is not affected much. Since only one tree might be impacted by the new data, it is difficult to impact all the trees. Moreover, random forest is comparatively less impacted by noise.

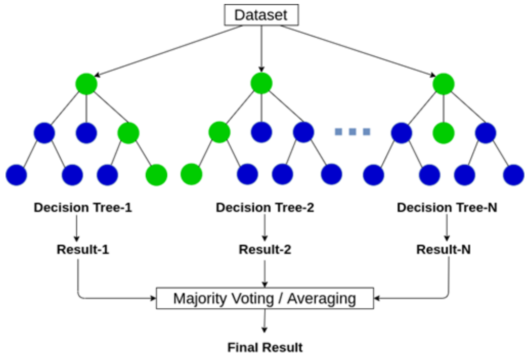


Fig. 2. How random forest algorithm works. (Source: [25])

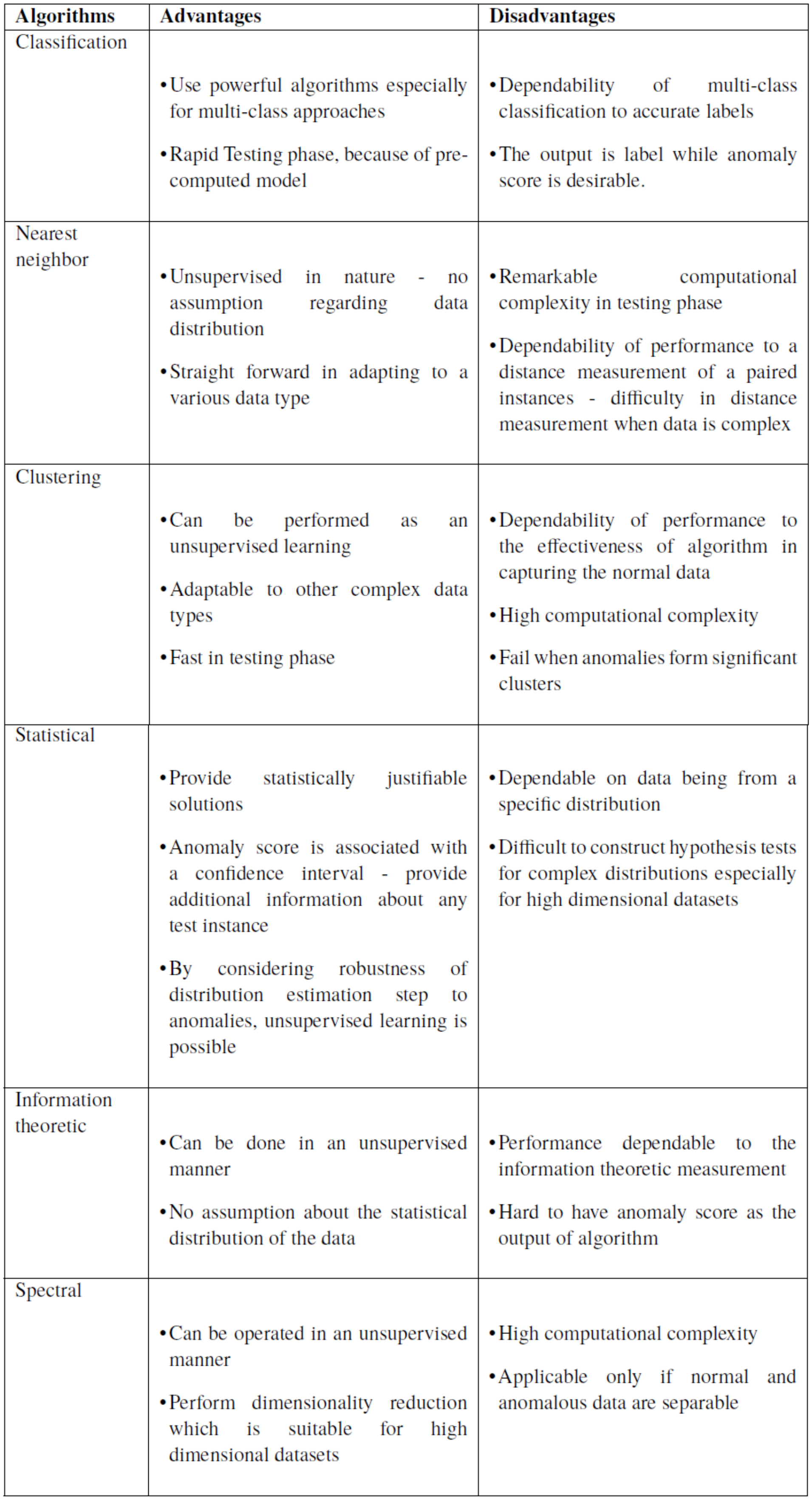
C. Ontology

Ontology is utilized to obtain knowledge about some domain of interest. An ontology defines the concepts in the domain and also the relationships that exist between these concepts, i.e., an ontology defines common words in order to share common understanding of the structure of information in a domain [26]. Various ontology languages provide different possibilities. Our focus is on introducing the components of OWL ontology as the most recent development in standard ontology languages [27]. An OWL ontology consists of Individuals, Properties, and Classes as the components. In the following, each of these components is introduced.

Individuals: Individuals expose objects in the domain of interest. OWL does not use the Unique Name Assumption (UNA). This implies that two different names could refer to the same individual. For instance, ’Queen Elizabeth’, ’The Queen’, and ’Elizabeth Windsor’, all of them might refer to the identical individual. In OWL, individuals must be explicitly declared that they refer to the same object or they are different. Figure 3 depicts a demonstration of some individuals in various domain.

Properties: Properties are relations that connect two individuals together. As shown in Figure 4, the property *livesIn* connects the individual *Matthew* to the individual *England*, or the property *hasSibling* links the individual *Matthew* to the individual *Gemma*. Properties could be inverted. For instance, the inverse of *hasOwener* property is *isOwnedBy* property. Also, properties could be either *transitive* or *symmetric*.

Classes: OWL classes behave like *sets* that contain individuals. They precisely declare the needs of the class memberships. For example, the class *Person* would contain all the individuals that are persons in the domain of interest. Classes might have superclass-subclass taxonomy. For instance, assume the classes *Animal* and *Dog* - *Dog* is the subclass of *Animal*. So, *Animal* is the superclass of *Dog*. This means that all dogs are animals



Algorithms	Advantages	Disadvantages
Classification	•Use powerful algorithms especially for multi-class approaches •Rapid Testing phase, because of pre-computed model	•Dependability of multi-class classification to accurate labels •The output is label while anomaly score is desirable.
Nearest neighbor	•Unsupervised in nature - no assumption regarding data distribution •Straight forward in adapting to a various data type	•Remarkable computational complexity in testing phase •Dependability of performance to a distance measurement of a paired instances - difficulty in distance measurement when data is complex
Clustering	•Can be performed as an unsupervised learning •Adaptable to other complex data types •Fast in testing phase	•Dependability of performance to the effectiveness of algorithm in capturing the normal data •High computational complexity •Fail when anomalies form significant clusters
Statistical	•Provide statistically justifiable solutions •Anomaly score is associated with a confidence interval - provide additional information about any test instance •By considering robustness of distribution estimation step to anomalies, unsupervised learning is possible	•Dependable on data being from a specific distribution •Difficult to construct hypothesis tests for complex distributions especially for high dimensional datasets
Information theoretic	•Can be done in an unsupervised manner •No assumption about the statistical distribution of the data	•Performance dependable to the information theoretic measurement •Hard to have anomaly score as the output of algorithm
Spectral	•Can be operated in an unsupervised manner •Perform dimensionality reduction which is suitable for high dimensional datasets	•High computational complexity •Applicable only if normal and anomalous data are separable

Fig. 1. Advantage and disadvantages of various anomaly detection techniques [9]

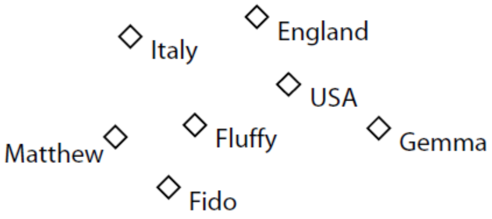


Fig. 3. Demonstration of individuals. (Source: [27])

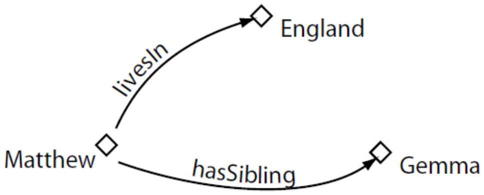


Fig. 4. Demonstration of properties. (Source: [27])

and all members of class *Dog* are members of class *Animal*. Figure 5 depicts a demonstration of some classes, which are containing some individuals. Classes are shown as circles or ovals and individuals are as the instances of classes.

IV. Data And Models For Experiments

This section sheds light on the data under investigation. Furthermore, it provides details on the pre-processing performed on the received data, and the planned data analytics and verification procedures.

A. Data Collection

The relation of data to AI is as food to the human being. In other words, there is no artificial intelligence in isolation, and any AI approach needs corresponding data for learning. For this project, we receive the dataset through our industrial partner, from a third-party company. While the data itself is confidential and could not be shared open access on the web, in this section we try to provide insights into the data, in order to make the reader familiar with the approaches that will be presented in the next section.

1) A deep Look into the Dataset: Our dataset contains 18 unique test runs for produced machine parts. Each of these tests has been run for a different period of time, i.e., there are different reported cycles per test.

2) Features available per test: The first dataset (testoverview.csv) provides a comprehensive list of features available per test (out of 18 tests). These features include the type of material used in the experiments, e.g., the oil, and the setting that has been applied in the experiment, e.g., distance between disks. This metadata has been collected to be used for verification of dataset and its reproducibility, as we will see in the next section (Section V-C).

3) Features available per test cycle: For each of the tests mentioned above, measurements have been done for different periods of time, and a number of features have been recorded per time cycle in the second dataset (tests.csv). In other words, this dataset presents a comprehensive list of features available per time cycle for each test. In contrast to the first dataset, most of the features of the second dataset are unknown to the reader and have not been revealed by the third company to us.

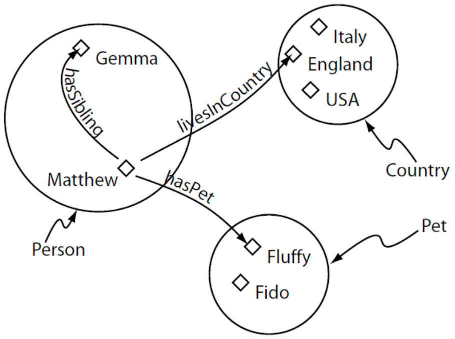


Fig. 5. Demonstration of classes, containing individuals and properties between them. (Source: [27])

B. Pre-processing of Data

For pre-processing of data, we investigate NaN values and missing entries in the dataset. Then, we start plotting the data to see trends in the results from each test. Figure 6 represents two features of a specific test across time. It is interesting to see that the features represent 3 trends in 3 different phases, including (a) an increasing trend at the start phase (up to 600 cycles, with a return to 50 periodically for the second feature), (b) a semi-constant trend from 600 cycles until the end cycle -600 cycles, and (c) an increasing trend in the last 600 cycles (with a return to 50 periodically for the second feature). In order to see if it is a recurring trend, we investigate the same thing for other tests. For example, Figure 7 represents the same phenomena for another test. Here, the start and end phases show a decreasing trend, while the middle phase is semi-constant with low variations. The increasing/decreasing trend at the start/end phases and the semi-constant trend in the middle phase are observed in all tests unless one test (depicted in Figure 8), and this test is excluded from our analysis based on the human expert information, as it does not show the standard behavior.

C. Planned Data Analysis

Figure 9 represents the plausibility check problem and the planned analysis for dealing with this problem. Based on this figure, we receive the data per test per time cycle (as the data pipeline from the bottom of the blue box), and also some metadata per test (as the left data pipeline), and aim at investigating if each test data is plausible or not. The focus of this work is on the design of the plausibility check module and the design of an ontology for the generation of the check data to be used in the plausibility checker module.

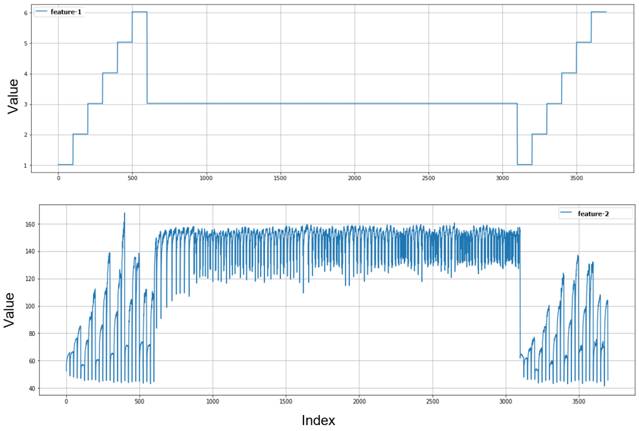


Fig. 6. Description of subset-1 of data versus cycle index

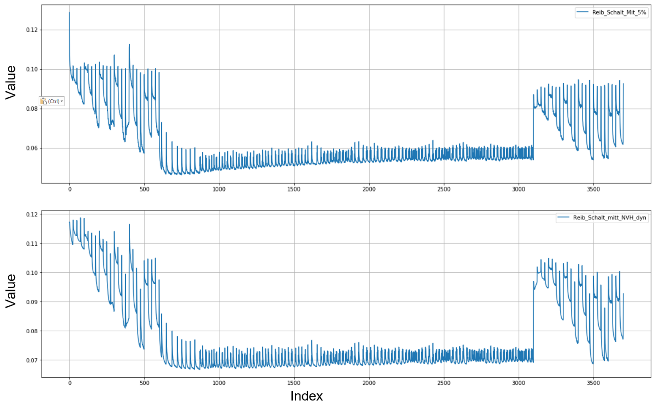


Fig. 7. Description of subset-2 of data versus cycle index

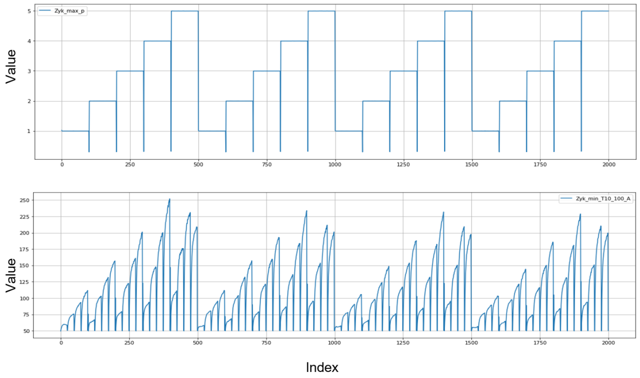


Fig. 8. Description of subset-3 of data versus cycle index

1) Evaluation Metric: In this work, we focus on predicting the test values and comparing them with the real values for detection of a potential anomaly, i.e., performing regression analysis. Regression refers to predictive modeling, and involves predicting a numeric value, and is different from the classification that involves predicting the label of a class of data. In regression analysis, we use Mean Squared Error (MSE), as an error metric designed for evaluating predictions made on regression problems. The MSE metric is derived as the mean or average of the squared differences between real and predicted values, i.e., $MSE = \frac{1}{N}\sum_{i=1}^{N}(X[i] - \tilde{X}[i])^2$, in which, $X[i]$ is the i'th real value in the dataset and $\tilde{X}[i]$ is the i'th predicted value. The difference is squared, which has the effect of resulting in a positive error value and inflating or magnifying the large errors.

2) Evaluation Framework: Figure 9 represents the evaluation framework for performance assessment of the proposed plausibility check solution. Based on this figure, we will add two types of error, including constant bias noise and random noise, to the test data per cycle, and will check if the plausibility check module is capable of finding inconsistency in the data.

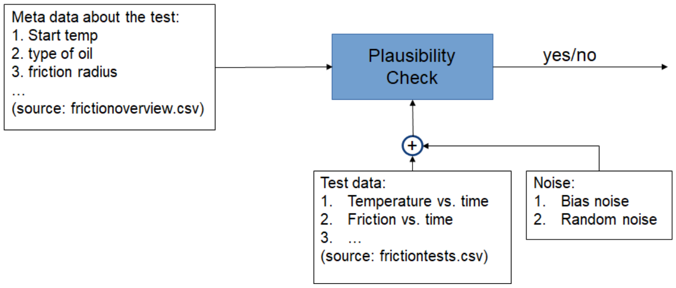


Fig. 9. Planned evaluation framework

V. The Proposed Solution

This section aims at presenting contributions of the work. Our contributions include the design of a data analytics unit for plausibility check of data. The schema of the proposed solution has been depicted in Figure 10. This proposed unit includes two novel functions: (a) the test data generator function and (b) the plausibility check function. The former one collects further information about the test and generates checkpoints (contextual data) to be evaluated by the checker function. The checker function compares the checkpoints with the threshold values and makes the plausibility decision. Then, before storing data in the database or actuating based on the received data, the customer can pass the data through the data analytics unit and check whether this data is plausible or not. As we will see in detail of the proposed approaches, the test data generator function includes an intelligent agent for generating the test data.

Implementation of the proposed solution requires contextual data to be collected. Contextual data is test data, which is

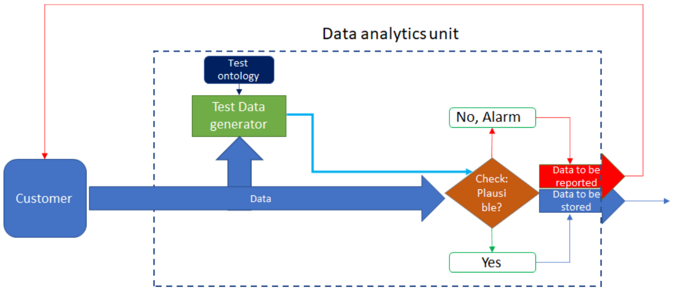


Fig. 10. The proposed solution

related to the dataset to be checked at the plausibility check function. Also, the contextual data should be contributory in the plausibility check of the dataset. In the following, three ideas are presented for generating contextual data:

1) Cross-correlation between columns of the dataset is used for prediction of the column of interest. The performance of prediction (MSE) is reported as a property of column of interest for a plausibility check.
2) Prediction of future values of each column based on the previous values of that column and comparison with the received data (Auto-regression). The performance in terms of MSE is used for a plausibility check.
3) Finding rules and statistics for each column based on metadata and configuration available for the test, e.g., type of oil used at the machine part.

A. Design of contextual information for plausibility check: The first solution

In tests.csv dataset, there are 18 unique tests with 29 data columns, unique hash codes, and different cycles. The columns of the dataset could be correlated together. Then, one can use some columns to check the plausibility of other columns.

For testing the hypothesis of mutual correlation between different columns, we consider one unique test and find the correlation between each column with itself and with 28 other columns, by using the built-in correlation function of Python. As shown in Figure 11, the correlation results of each test are stored in a matrix of $29 * 29$. The correlation number in each cell $c_{i,j}$ of this matrix is an amount between -1 and 1 and this number states that how much the column i is correlated to the column j. The higher the absolute value of each cell $c_{i,j}$, the more correlated the column i to the column j.

Since the correlations between columns in one test might randomly be high or low, the correlation matrix is calculated for each 18 unique tests, and 18 correlation matrices of $29 * 29$ are obtained. Then, each cell of correlation matrices is averaged over all 18 tests. Figure 12 refers to the result of averaged correlation matrices over 18 tests. This correlation matrix is for the starting phase. Since the behavior of features in the various phases is different, the correlation matrix for the steady-state and ending phase are calculated separately.

As the absolute value of the correlation matrix is of importance, the features with the hottest and coldest colors are more correlated together. As shown in Figure 12, the results confirm the existence of strongly related features for plausibility check of each feature.

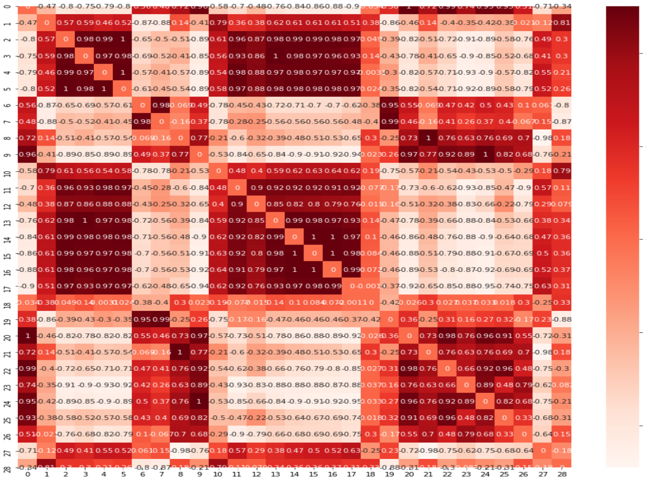


Fig. 11. The correlation matrix for one test before averaging

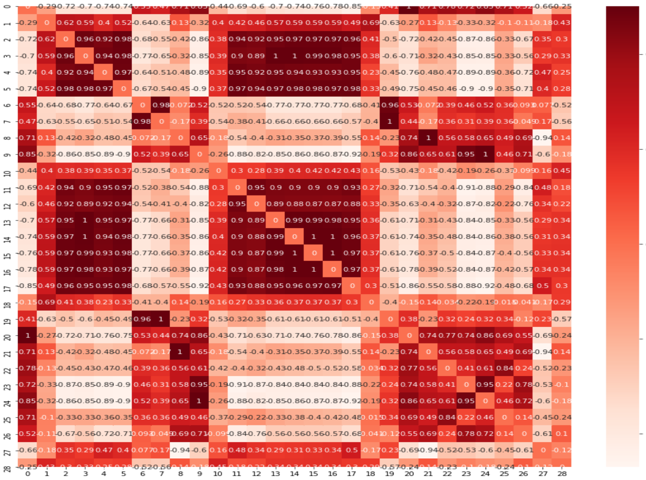


Fig. 12. The correlation matrix after averaging over all available tests

Having access to the m most related columns for each column, we can train a machine-learning algorithm to predict the value of feature of interest (FoI) based on the selected features. Here, we select the three most related features for prediction. If the prediction based on the selected features matches the recorded data, there is a low probability of implausibility. If the predicted and recorded values do not match, an alarm could be raised. For deploying this idea, we need an ML agent. Figure 13 depicts the check data generation and decision-making procedures in more detail. In this figure, the FoI is X_1, and the subset of features related to it is X_2. Then, X_2 is fed to the test generator node, and a prediction of X_1 based on X_2 is generated (call it $\tilde{X}_1$). The predicted value, $\tilde{X}_1$ along with X_1 are fed to the comparator

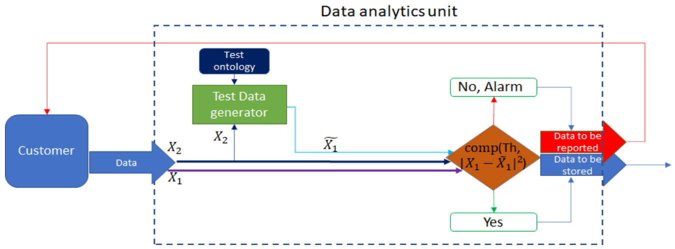


Fig. 13. Feature selection and decision making procedure in more details for the first solution.

node, and from the comparison, the system can carry out the validation process. Finally, the X_1 data will be accepted or an alarm will be triggered. One must note that the test ontology can trigger generating any kind of test data for X_1 based on X_2. For example, after setting the ontology by a human expert, the ML agent in the test generator node takes ontology and customer dataset as inputs. Ontology determines what contextual information should be collected. In the above example, ML agent understands from the ontology that MSE is required to be collected for the FoI. So, the ML agent, by applying an appropriate algorithm, generates the MSE in the prediction of feature-1 using the three most related features to it. In the decision-making step, this MSE is compared with the ground-truth value. If the value of MSE is less than or equal to the ground-truth value, then the customer data is plausible and can be stored in the database, otherwise, the data is implausible and an alarm is raised.

Towards deploying the ML agent, we need to select an ML algorithm, prepare a train and test dataset, train it over train dataset, and test it over test dataset. To select an ML algorithm, we need to consider some points such as simplicity in usage, scalability, being model-free, explainability, resistance against overfitting and noise, resistance against non-available values in measurements, and working with categorical and continuous values. Regarding these tips, a random forest (RF) algorithm for regression is selected to be implemented in the ML agent. Investigation of the RF algorithm on our dataset for configuration of its parameter, i.e., number of estimator trees, showed us that the best performance, in terms of speed and overfitting, is achieved by 50 trees. Performance of the RF algorithms for plausibility check is investigated in subsection VI-A of the next section. Towards using RF algorithm, we train an RF agent based on several tests (out of 18 as described in the previous section), and then test this agent on a test dataset (excluding the training datasets).

B. Design of contextual information for plausibility check: The second solution

Not only does cross-correlation exists between columns of the dataset, but also auto-correlation among values of one column could be considered. It means that one can utilize the previous values of a column to check the plausibility of a specific value in this column.

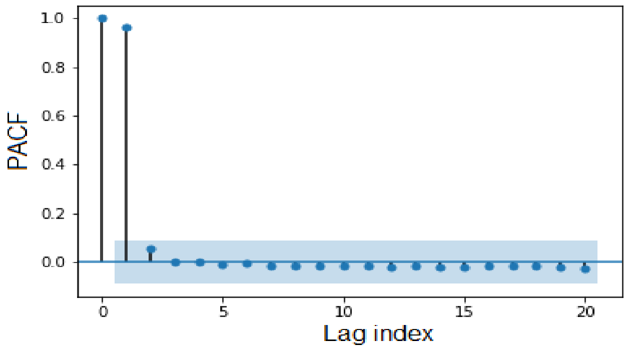


Fig. 14. Auto-correlation of feature 1 in the starting phase of test A

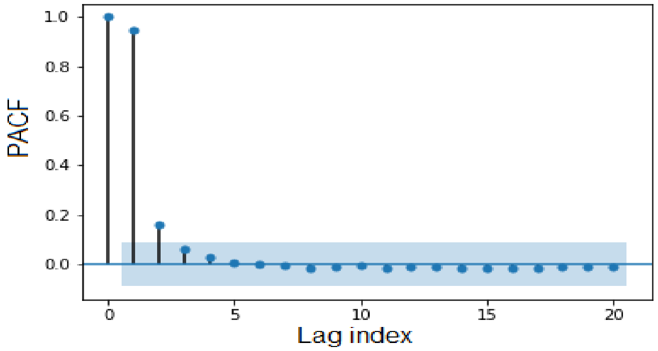


Fig. 15. Auto-correlation of feature 1 in the starting phase of test B

To see if auto-correlation could be used for the prediction of a feature from its lags, we consider one unique test and find the auto-correlation for each feature of this test. By using auto-correlation, we can find how a value of a feature in time t is related to the previous values of this feature at time $t-1$, $t-2$, $t-3$, ..., $t-n$. Figure 14 and Figure 15 depict the auto-correlation of E-spec in starting phase of test A and B respectively.

Since the values of auto-correlation for a specific feature of one test could randomly be high or low, we repeat auto-correlation for this feature over the 18 tests and average the

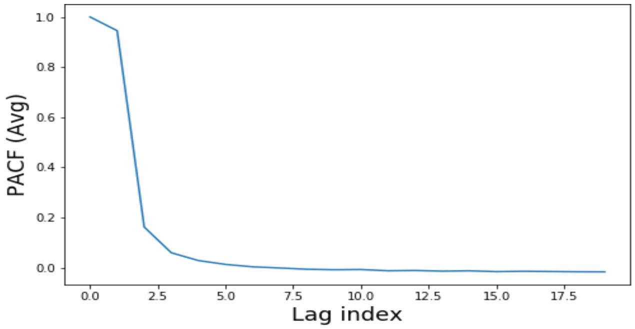


Fig. 16. Average of auto-correlation for feature 1 over different tests in the starting phase

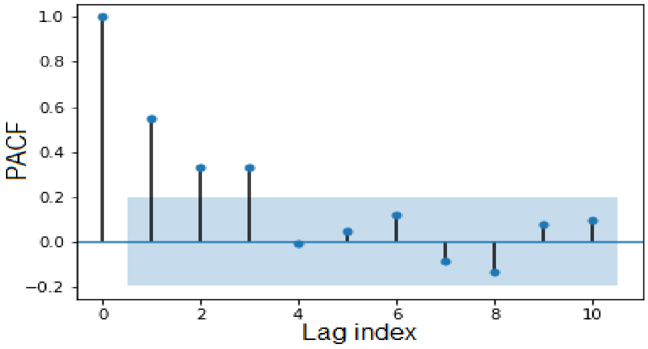


Fig. 17. Auto-correlation of feature 2 in the steady phase of test C

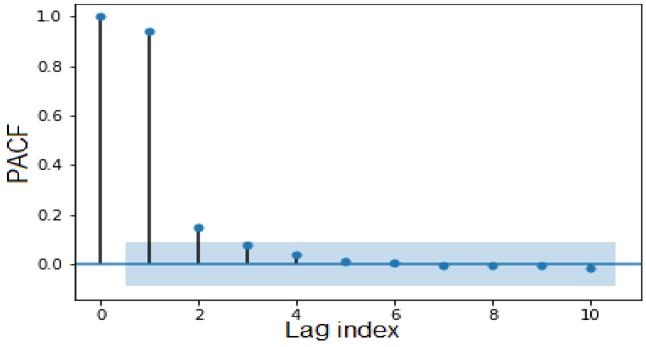


Fig. 18. Auto-correlation of feature 1 in the end phase of test C

values of these tests. So, Figure 16 is resulted. Then, among these averaged values, previous m recent values are selected for use in the ML agent. Since the behavior of features in the various phases follows different models, the auto-correlation function is calculated for each phase of a feature separately. Figure 17 and Figure 18 refer to the auto-correlation functions in the steady phase and end phase of feature 1 and 2 for the same test. Figure 19 and Figure 20 show the average of auto-correlation function over different tests in the steady phase of feature 2 and ending phase of feature 1.

Having access to the previous m recent values of a feature, we can train a machine-learning algorithm to predict the value of FoI at time t based on the previous values of the feature at time $t-1$, $t-2$, ..., and $t-n$. If the prediction value at time t based on the previous m recent values match the recorded data, there is low probability of implausibility, otherwise because of mismatch of prediction data and recorded one, an alarm could be raised. Figure 21 depicts the overall architecture of the second solution in more detail. In this figure, part of $X_1[0:N_2]$, e.g., $X_1[0:N_1]$ in which $N_1 < N_2$, is fed to the test data generator (Note: X_1 is the FoI.). Then, based on the test ontology, e.g., time series forecasting of X_1 using ARIMA, test data for the validity of $X_1[0:N_2]$ will be generated, e.g., $\tilde{X}_1$. Finally, at the comparator node, the real value of X_1 will be compared against $\tilde{X}_1$. Based on this comparison, X_1 data will be accepted or an alarm will be triggered.

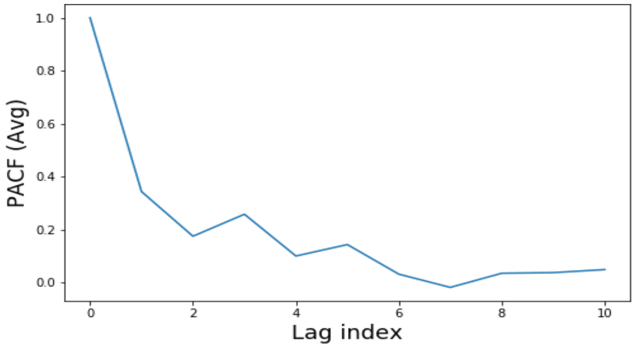


Fig. 19. Average of auto-correlation for feature 2 over different tests in the steady phase

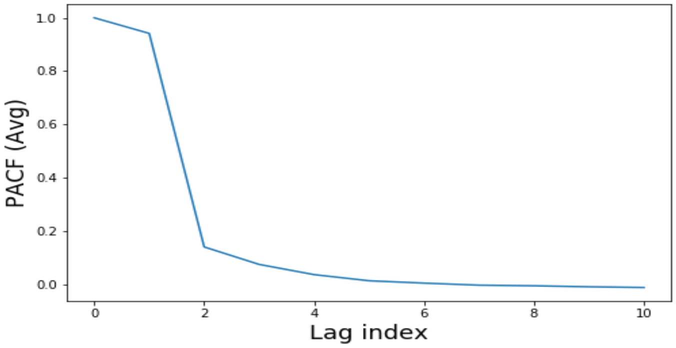


Fig. 20. Average of auto-correlation for feature 1 over different tests in the end phase

Toward deploying an ML agent for the second hypothesis, Random Forest (RF) and ARIMA algorithms are implemented. As mentioned in subsection V-A, we use the RF algorithm with 50 estimators for our test purpose. For the RF algorithm, the plausibility of each data point is checked based on the 10 lags of the data, i.e., $x[n]$ is checked based on $x[n\text{-}10]$:$x[n\text{-}1]$. For ease of notation, we call this RF algorithm as RF(50,10). For the ARIMA approach, the investigation of parameters on our dataset showed that P=3, Q=I=0, i.e., ARIMA(3,0,0) matches our dataset. Performance of ARIMA and RF algorithms for

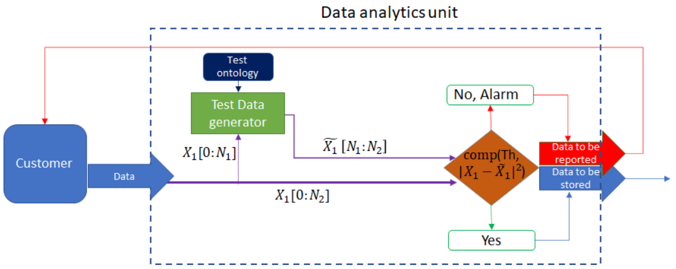


Fig. 21. Feature selection and decision making procedure in more details for the second solution.

plausibility check is investigated in subsection VI-B of the next section. Towards using ARIMA and RF algorithms, we train the ML agent based on several datasets (out of 18 tests), and then test these agents on a test dataset (excluding training tests).

C. Design of contextual information for plausibility check: The third solution

In the previous sections, we have leveraged the information in the features, either in the FoI or a combination of features, for plausibility check. In other words, the other contextual data gathered by the test maker related to the overall test have not been considered. In this section, we aim at investigating the impact of such contextual data on the statistics of FoI, and the potential application of such connection in plausibility check for the dataset. Figure 22 represents the overall structure of the proposed solution. In this figure, the metadata about X_1, which is the FoI, is fed to the test data generator along with X_1. Then, based on the test ontology, e.g., partitioning Cumulative Distribution Function (CDF) of X_1 based on states of the metadata, test data for the validity of X_1 will be generated, e.g., $\tilde{S}_{X_1}$. Finally, at the comparator node, the real value of S_{X_1} from received X_1, e.g., the average value of X_1 will be compared against the $\tilde{S}_{X_1}$. Based on this comparison, X_1 data will be accepted or an alarm will be triggered.

In our dataset, there are several contextual information corresponding to each unique test that potentially have impacts on the statistics of features. Examples of such contextual data include type of the *oil* and *separator metal* used in the experiment. Let us focus on oil. The initial hypothesis is that there is a connection between the type of oil used in a test and the statistics of measurements in this test. For example, the min, max, variance, median, mean values of distribution for Oil-A have considerable differences from the ones of Oil-B. Figure 23 and Figure 24 show the statistics for *feature-2*. One can observe that the Probability Density Function (PDF) and Cumulative Distribution Function (CDF) of this feature are different for various oil types. Furthermore, the min and max values of this feature for *type-A* oil differ from *type-B* oil. So, using these explored statistics, we can add some rules to the ontology to discover the implausibility of the data. If the data would be implausible, the related statistics will change in comparison with the normal ones.

We train the metrics of decision-making using statistics of feature-2. If statistics of the test dataset comply with the statistics of the trained dataset, i.e., metrics like min, median, and variance are within the accepted bound found in the training, the decision-maker accepts the test data as plausible. Performance of plausibility check by using statistics of the data is investigated in subsection VI-C of the next section.

VI. Results And Discussion

A. Performance test for the first solution

Recall the first proposed solution in Figure 13. In this solution, the test ontology mandates predicting FoI for validity check based on the three most-related features. It also proposes MSE as the prediction analysis metric. Then, the three most related features to the FoI are fed to the test data generator and are used for predicting the FoI. Figure 25 shows the performance test results such that the prediction values are fitted well with the real values of FoI (here, feature-1).

In this figure, along with the test data and predicted data, the three most related features to feature-1 can be seen as well. One can observe that these three features have almost either direct or inverse (because of negative values of correlation) relationship with the feature of interest (feature-1). The above tests have been repeated for the steady phase and ending phase, and the same behavior has been almost observed for tests in these phases. We aim at leveraging the proposed ML agent

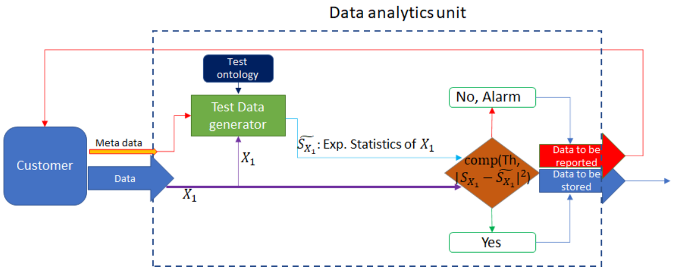


Fig. 22. Feature selection and decision making procedure in more details for the third solution.

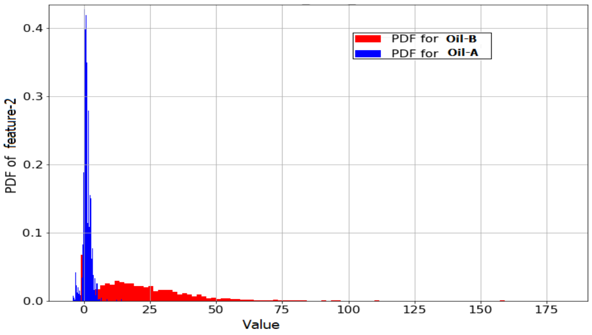


Fig. 23. PDF of *feature-2*

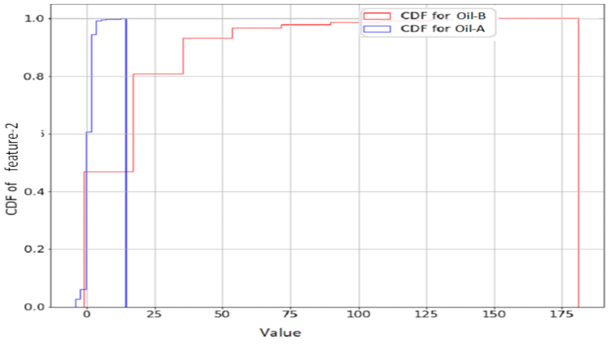


Fig. 24. CDF of feature-2

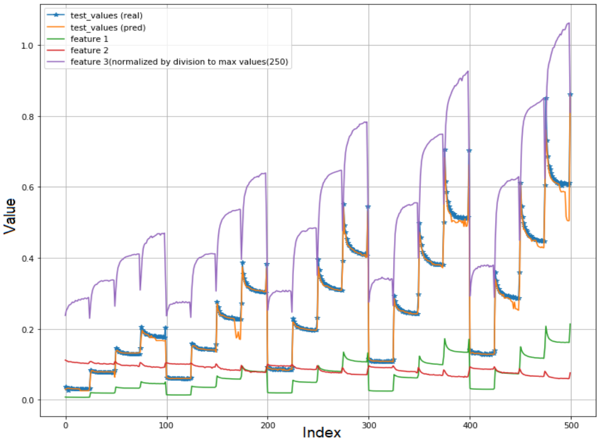


Fig. 25. Testing agent for predicting feature-1 with more details of 3 most related features

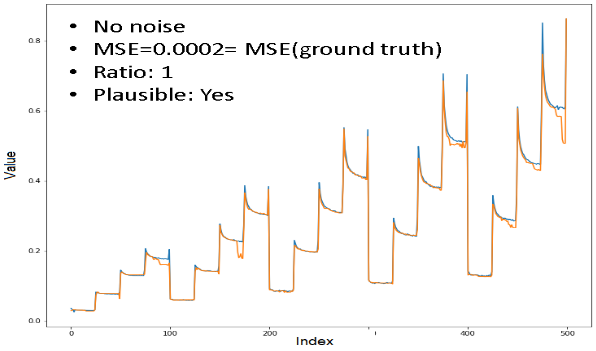


Fig. 26. Testing agent for predicting feature-1 based on 3 most related feature. No noise has been applied, MSE=0.0002= MSE (ground truth). (Blue: real data, Orange: predicted data)

for carrying plausibility checks out. So, seven test cases are presented based on applying bias measurement errors, and random measurement errors to the column of interest, and most related columns. The first plausibility check is related to the state that there is no noise in the data. As shown in Figure 26, plausibility of data has been confirmed. In second plausibility test, bias noise is added on the feature of interest (E-spec). From results of Figure 27, it can be observed that the predicted values are not the same as real values. So, the ML agent can detect the error on the data and conclude the implausibility of data. The third plausibility test, as shown in Figure 28, is related to the adding bias noise to the least related feature. In forth plausibility check, bias noise is added to the most related feature. The result is depicted in Figure 29. The same plausibility tests are done by adding random noise on the feature of interest, least related feature, and most related feature. The results of these tests are shown in Figure 30, Figure 31, and Figure 32, respectively.

B. Performance test for the second solution

Recall the second proposed solution in Figure 21. In this solution, the test ontology mandates predicting FoI for validity check based on its lags. It also proposes MSE as the prediction

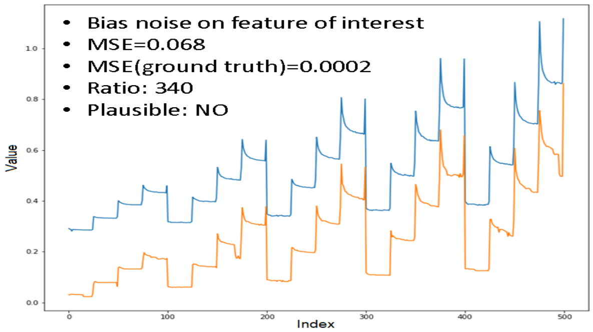


Fig. 27. Testing agent for predicting feature-1 based on 3 most related features. Bias noise on the feature of interest, MSE=0.068 (340 times more than ground truth). (Blue: real data, Orange: predicted data)

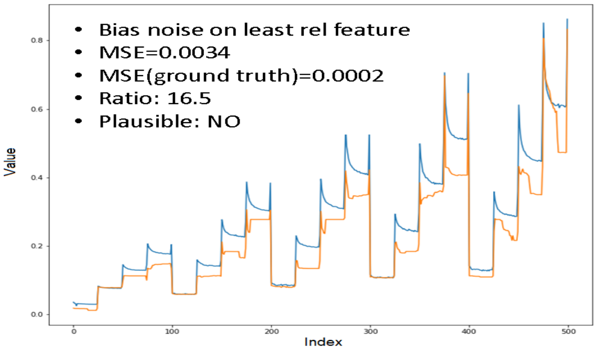


Fig. 28. Testing agent for predicting feature-1 based on 3 most related feature. Bias noise on the least related feature, MSE=0.0034 (16.5 times more than the ground truth). (Blue: real data, Orange: predicted data)

analysis metric. Then, the lags of FoI are fed to the test data generator, and are used for predicting the FoI. Figure 33 shows the performance test result using random forest for predicting feature-1 (FoI). In the random forest algorithm, we used 10 recent values of feature-1 for prediction. Figure 34 depicts the performance test result for predicting feature-1 using auto-correlation and ARIMA. In our implementation, ARIMA works with three recent values of feature-1. Both Figure 33 and Figure 34 confirm that the prediction values fit well with the real values of feature-1. We do the performance

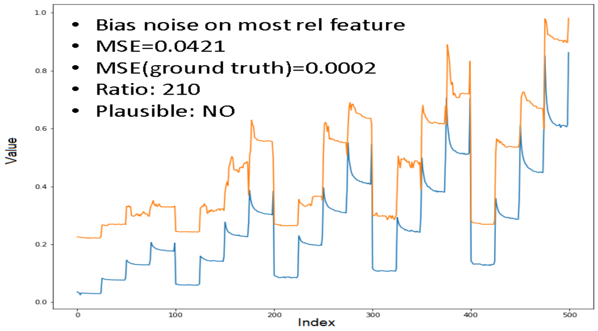


Fig. 29. Testing agent for predicting feature-1 based on 3 most related feature. Bias noise on the most related feature, MSE=0.0421 (210 times more than the ground truth). (Blue: real data, Orange: predicted data)

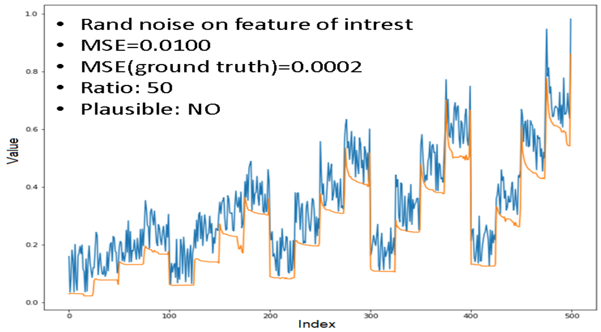


Fig. 30. Testing agent for predicting feature-1 based on 3 most related feature. Random noise on the feature of interest, MSE =0.01 (50 times higher than the ground truth). (Blue: real data, Orange: predicted data)

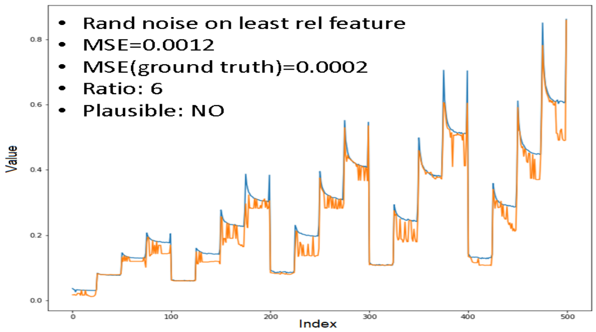


Fig. 31. Testing agent for predicting E-spec based on 3 most related feature. Random noise on the least related feature, MSE=0.0012 (6 times more than the ground truth).

test for the steady phase and ending phase of feature-1 using random forest and ARIMA algorithms and the results for these phases also follow the same trend.

For plausibility check using the auto-correlation contextual data, we apply bias measurement errors and random measurement errors to the FoI, and examine if the proposed solution can assess the incorrectness of data. Towards this end, we leverage the FoI's forecasting results using ARIMA and random forest methods. From Figure 35, Figure 36, Figure 37, and Figure 38 with random and bias noises, one can observe that the predicted values are not the same as real values of FoI. So, the ML agent can detect the error on the data and conclude the implausibility of the data. Table II summarizes the results of plausibility tests for the second solution.

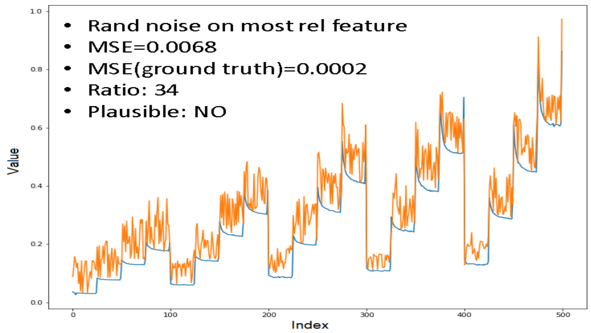


Fig. 32. Testing agent for predicting feature-1 based on 3 most related feature. Random noise on the most related feature, MSE= 0.00068 (34 times more than the ground truth). (Blue: real data, Orange: predicted data)

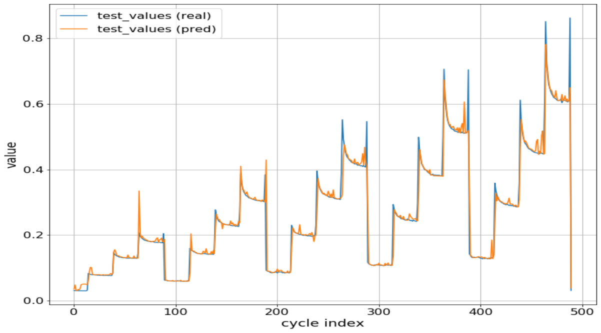


Fig. 33. Testing agent for predicting feature-1 using auto-correlation and random forest.

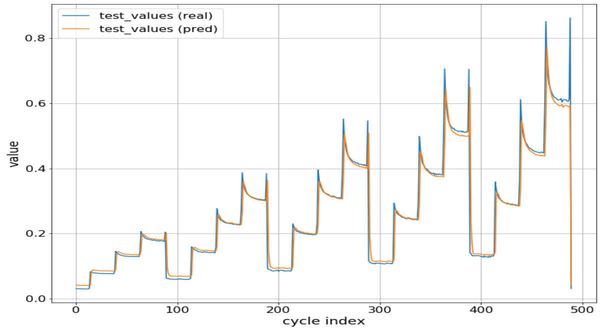


Fig. 34. Testing agent for predicting feature-1 using auto-correlation and ARIMA.

C. Performance test for the third solution

Recall the third proposed solution in Figure 22. In this solution, the test ontology collects metadata about FoI for validity check. Then, the past values of this feature are fed to the test data generator, and are used for extraction of statistics of this feature, and predicting the validity of the feature based on the extracted statistics. Here, we focus on the oil data and try to partition the PDF of FoI based on the type of oil used

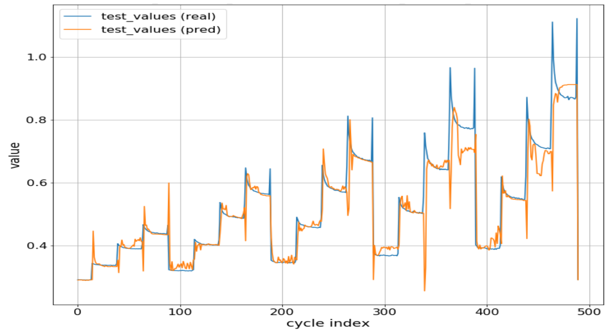


Fig. 35. Plausibility check for predicting feature-1 using auto-correlation, Random forest, and bias noise.

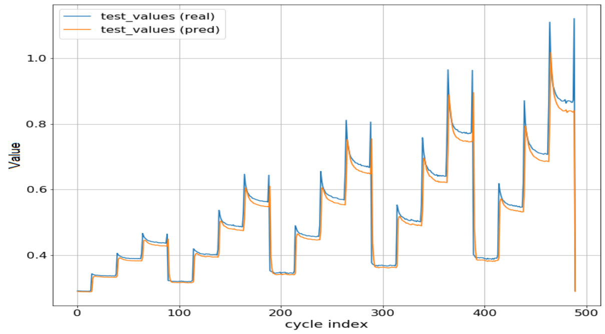


Fig. 36. Plausibility check for predicting feature-1 using auto-correlation, ARIMA, and bias noise.

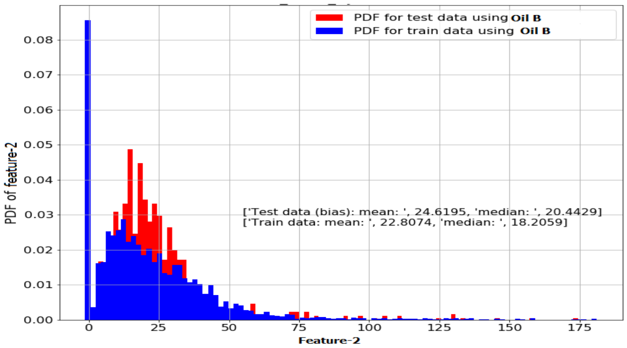


Fig. 39. Comparison of PDF of FoI in two tests

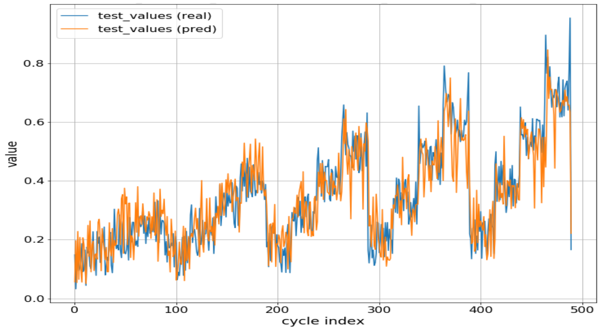


Fig. 37. Plausibility check for predicting feature-1 using auto-correlation, random forest, and random noise.

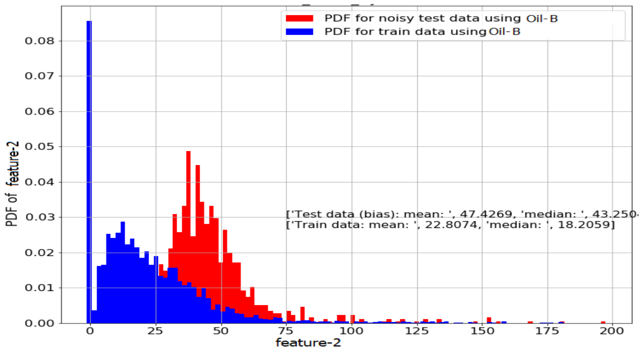


Fig. 40. Comparison of PDF of FoI with and w/o bias noise

in the experiment. Figure 39 represents the partitioned PDF of the feature-2 (FoI) based on the type of oil used in the experiment. One observes the same trend from the test data and train data when there is no noise added to data (plausible test dataset).

In this section, we apply bias noise and random noise on the test data to check if our designed solution can detect the implausible data. Figure 40 and Figure 41 show the results of performance analysis for bias and random noise respectively. One observes in Figure 40 that adding the noise to the test data (red one) clearly shifts the plot to the right. Figure 41 represents the dataset with random noise. One can observe that the noise added to the data has changed the shape of PDF in both cases of bias and random noise, e.g., the mean and median have changed in Figure 40 and Figure 41 in comparison with the original data without noise shown in Figure 39.

D. Discussion

In Table I, the results of plausibility test for the first solution (subsection VI-B) have been summarized in more details. Table II summarizes the performance results for the second solution (subsection VI-B). Table III summarizes the results of Figures 39, 40, and 41 in subsection VI-B.

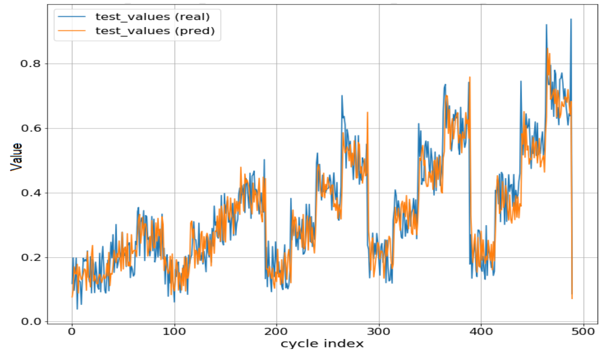


Fig. 38. Plausibility check for predicting feature-1 using auto-correlation, ARIMA, and random noise.

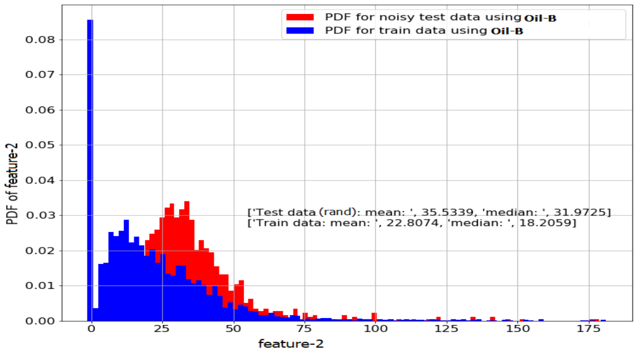


Fig. 41. Comparison of PDF of FoI with and w/o random noise

TABLE I
SUMMARY OF PLAUSIBILITY CHECK USING SOLUTION 1

Test description	**MSE**	**MSE**$_{\text{ratio}}$: $\frac{\text{MSE}}{\text{MSE}_{\text{true}}}$	**Check**: MSE$_{\text{ratio}}$ $<$ Ratio$_{\text{th}}$; Ratio$_{\text{th}}$ $=$ 1.5
True data	0.0002	1	Y
Bias error on column of interest (feature-1)	0.068	360	N
Bias error on least related feature	0033	16.5	N
Bias error on most related feature	0.0420	210	N
Random error on feature of interest (feature-1)	0.010	50	N
Random error on least related feature	0.0012	6	N
Random error on most related feature	0.0068	34	N

TABLE III
SUMMARY OF PLAUSIBILITY CHECK USING SOLUTION 3

Test description	**Mean**	**Mean-ratio**	**Median**	**Median -ratio**	**Plaus. check**
Train data (base measurement)	22.8	1	18.2	1	-
Test data w/o noise	24.6	1.08	20.4	1.12	Y
Test data with bias error	47.42	2.08	43.4	2.38	N
Test data with random error	35.8	1.57	31.7	1.74	N

From Table I, it is clear that the plausibility check solution, which is powered by the prediction of FoI based on the most related features, performs well against the bias noise. In other words, when a constant value, i.e., a measurement error, is added to the reading of a sensor, the plausibility check module can easily detect that data is inconsistent with the past learning (from 16.5 to 360 times more MSE has been reported). For the random noise, when the amount of the added noise to the data could vary, the performance is lower than the bias noise, but still completely acceptable (from 6 to 50 times more MSE has been reported). For example, one observes that the plausibility test has been shown 6 times more MSE in the prediction of FoI when random noise on the least relevant feature to the FoI has been added. Furthermore, Table II showed that the second solution (using RF) is not vulnerable to the random noise, and it performs equivalently for the bias and random noises (7 times more MSE in prediction of FoI). In the same time, we observe that the ARIMA has a poor performance as an ML agent for this solution, and it misses the alarm for the test-case with bias noise on the the FoI (the corresponding MSE-ratio is 1.125, which is lower than the threshold value, i.e., 1.5). Finally, the third approach shows a weaker performance than the previous ones (around two times more MSE has been reported). One must note that the stronger performance of the first approach and relatively the second approach is achieved at the cost of further computing required for them. In other words, there is a hidden reliability-complexity trade-off here, where going from solution 1 to 3, complexity is reduced and the probability of error in plausibility check is increased.

VII. Conclusions

In this work, we investigated data plausibility automation for a given dataset from a smart factory. Towards this end, a data analytics framework, consisting of a contextual data generation function (which generates checkpoints based on a given ontology) and a plausibility check function (which works based on the designed checkpoints), was proposed. For the implementation of the first function, we have investigated three machine learning approaches that leverage auto-correlation in each feature, correlation between features, and hidden statistics of each feature for generating the checkpoints. Performance evaluation results indicated the outstanding performance of the proposed schemes in the detection of noisy data. The main concluding remarks of this work include: (i) This study indicated that each feature of the dataset, or a collection of features, could be used without any other data for plausibility check leveraging machine learning. (ii) Metadata about the test, including conditions in which the test has been carried out, could be an important part of the design of the plausibility check. (iii) Checking of plausibility for a dataset that may contain random noise on some features (or some cycles) is much harder than checking the presence of static noise on the data. (iv) Performances of different checkpoint generation functions (using different ML approaches) are not the same. The ones based on the investigation of each cycle of the test, solutions 1 and 2, are more complex and provide a better distinction between noisy and healthy data. While the third solution is a lightweight solution with a lower reliability performance.

Acknowledgement

The research reported in this paper has been partially funded by BMK, BMDW, the State of Upper Austria in the frame of the COMET Programme managed by FFG and by the EC H2020 project *"DataCloud: Enabling the Big Data Pipeline Lifecycle on the Computing Continuum"* (Grant nr. 101016835).

References

[1] M. Basiri, J. Himmelbauer, L. Ehrlinger, and M. Matskin, "Context-aware data plausibility check using machine learning," in *The Fourteenth International Conference on Advances in Databases, Knowledge, and Data Applications*. DBKDA, 2022.

[2] V. Q. Nguyen, L. Van Ma, and J. Kim, "Lstm-based anomaly detection on big data for smart factory monitoring," *Journal of Digital Contents Society*, vol. 19, no. 4, pp. 789–799, 2018.

[3] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 2015, pp. 1939–1947.

[4] S. So, J. Petit, and D. Starobinski, "Physical layer plausibility checks for misbehavior detection in v2x networks," in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*, 2019, pp. 84–93.

[5] C. C. Aggarwal, "Outlier analysis," in *Data mining*. Springer, 2015, pp. 237–263.

TABLE II
SUMMARY OF PLAUSIBILITY CHECK USING SOLUTION 2

Test description	**MSE in prediction of FoI using itself by ARIMA**	**$\mathbf{MSE_{ratio}}$ for ARIMA: $\frac{\mathbf{MSE}}{\mathbf{MSE_{true\text{-}AR}}}$**	**Plausibility for ARIMA:** $\mathrm{MSE_{ratio}} < \mathrm{Ratio_{th}}$; $\mathrm{Ratio_{th}} = 1.5$	**MSE in prediction of FoI using itself by RF**	**$\mathbf{MSE_{ratio}}$ for RF: $\frac{\mathbf{MSE}}{\mathbf{MSE_{true\text{-}RF}}}$**	**Plausibility for RF: $\mathbf{MSE_{ratio}} < \mathbf{Ratio_{th}}$; $\mathbf{Ratio_{th}} = 1.5$**
True data (feature-1)	$0.0024 = \mathrm{MSE_{true\text{-}AR}}$	1	Y	$0.0012 = \mathrm{MSE_{true\text{-}RF}}$	1	Y
Bias error on feature of interest (feature-1)	0.0027	1.125	Y	0.0046	7.8	N
Random error on feature of interest (feature-1)	0.0063	2.8	N	0.0088	7.3	N

[6] C. C. Aggarwal and S. Sathe, *Outlier ensembles: An introduction*. Springer, 2017.

[7] S. Thudumu, P. Branch, J. Jin, and J. J. Singh, “A comprehensive survey of anomaly detection techniques for high dimensional big data,” *Journal of Big Data*, vol. 7, no. 1, pp. 1–30, 2020.

[8] L. Ehrlinger and W. Wöß, “Towards a definition of knowledge graphs.” *SEMANTiCS (Posters, Demos, SuCCESS)*, vol. 48, pp. 1–4, 2016.

[9] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[10] V. Roth, “Kernel fisher discriminants for outlier detection,” *Neural computation*, vol. 18, no. 4, pp. 942–960, 2006.

[11] D. Barbara, N. Wu, and S. Jajodia, “Detecting novel network intrusions using bayes estimators,” in *Proceedings of the 2001 SIAM International Conference on Data Mining*. SIAM, 2001, pp. 1–17.

[12] Y. Song, J. Huang, D. Zhou, H. Zha, and C. L. Giles, “Iknn: Informative k-nearest neighbor pattern classification,” in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2007, pp. 248–264.

[13] A. H. Abuzaid, “Identifying density-based local outliers in medical multivariate circular data,” *Statistics in Medicine*, vol. 39, no. 21, pp. 2793–2798, 2020.

[14] G. Moreira, M. Y. Santos, J. M. Pires, and J. Galvão, “Understanding the snn input parameters and how they affect the clustering results,” *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 11, no. 3, pp. 26–48, 2015.

[15] R. Smith, A. Bivens, M. Embrechts, C. Palagiri, and B. Szymanski, “Clustering approaches for anomaly based intrusion detection,” *Proceedings of intelligent engineering systems through artificial neural networks*, vol. 9, 2002.

[16] S. Ali, G. Wang, R. L. Cottrell, and T. Anwar, “Detecting anomalies from end-to-end internet performance measurements (pinger) using cluster based local outlier factor,” in *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*. IEEE, 2017, pp. 982–989.

[17] F. W. Scholz, “Maximum likelihood estimation,” *Wiley StatsRef: Statistics Reference Online*, 2014.

[18] S. Siami-Namini, N. Tavakoli, and A. S. Namin, “A comparison of arima and lstm in forecasting time series,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 1394–1401.

[19] G. E. Box and G. C. Tiao, “A bayesian approach to some outlier problems,” *Biometrika*, vol. 55, no. 1, pp. 119–129, 1968.

[20] Y. Xiao, C. Xing, T. Zhang, and Z. Zhao, “An intrusion detection model based on feature reduction and convolutional neural networks,” *IEEE Access*, vol. 7, pp. 42 210–42 219, 2019.

[21] E. Parzen, “On estimation of a probability density function and mode,” *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[22] P. Vitányi, “How incomputable is kolmogorov complexity?” *Entropy*, vol. 22, no. 4, p. 408, 2020.

[23] L. Parra, G. Deco, and S. Miesbach, “Statistical independence and novelty detection with information preserving nonlinear maps,” *Neural Computation*, vol. 8, no. 2, pp. 260–269, 1996.

[24] A. Azari, P. Papapetrou, S. Denic, and G. Peters, “User traffic prediction for proactive resource management: learning-powered approaches,” in *2019 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[25] A. Sharma. (2020) Decision tree vs. random forest – which algorithm should you use? Accessed: 2022-12-15. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm

[26] N. F. Noy, D. L. McGuinness *et al.*, “Ontology development 101: A guide to creating your first ontology,” 2001.

[27] M. Horridge, S. Jupp, G. Moulton, A. Rector, R. Stevens, and C. Wroe, “A practical guide to building owl ontologies using protégé 4 and co-ode tools edition1. 2,” *The university of Manchester*, vol. 107, 2009.