November 17, 2022
Singapore, Singapore

1947 75 2022

Association for
Computing Machinery

SEA4DQ '22

Proceedings of the 2nd International Workshop on

Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things

Edited by:

Phu Nguyen, Sagar Sen, and Maria Chiara Magnanini

Sponsored by:

ACM SIGSOFT, National University of Singapore

Co-located with:

ESEC/FSE '22

Association for Computing Machinery, Inc.
1601 Broadway, 10th Floor
New York, NY 10019-7434
USA

ACM ISBN: 978-1-4503-9459-8

Cover photo:
Title: "Helix Bridge and Marina Bay Sands"
Photographer: Erwin Soo
License: Creative Commons Attribution 2.0 Generic
https://creativecommons.org/licenses/by/2.0/deed.en
Cropped from original:
https://commons.wikimedia.org/wiki/File:Helix_Bridge_and_Marina_Bay_Sands_(8061798457).jpg

Production: Conference Publishing Consulting
D-94034 Passau, Germany, info@conference-publishing.com

Message from the Chairs

Welcome to the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (SEA4DQ 2022), November 17th, 2022, co-located with the ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC / FSE) 2022, Singapore.

Cyber-physical systems (CPS)/Internet of Things (IoT) are omnipresent in many industrial sectors and application domains in which the quality of the data acquired and used for decision support is a common factor. Data quality can deteriorate due to factors such as sensor faults and failures due to operating in harsh and uncertain environments. *How can software engineering and artificial intelligence (AI) help manage and tame data quality issues in CPS/IoT?* In this workshop, we aim to answer this question.

Data quality is of paramount importance for CPS/IoT. This workshop series stemmed from the common interest in data quality of the Zero-Defect Manufacturing (ZDM) Research and Innovation projects under the Horizon 2020 Framework Programme such as InterQ (https://interq-project.eu/) and DAT4.Zero (https://dat4zero.eu/). Not only for ZDM, but also in general, emerging trends in software engineering need to take data quality management seriously as CPS/IoT are increasingly data-centric in their approach to acquiring and processing data along the edge-fog-cloud continuum. This workshop provides researchers and practitioners a forum for exchanging ideas, experiences, understanding of the problems, visions for the future, and promising solutions to the problems in data quality in CPS/IoT.

Compared to the first edition SEA4DQ 2021, which featured one keynote, five presentations, and one panel discussion, the second edition SEA4DQ 2022 has evolved significantly with two keynotes, eight paper submissions, six presentations, and one panel discussion. Eight papers submitted to SEA4DQ 2022 had gone through a rigorous review process by the Program Committee, with three/four reviewers per paper. Submissions of PC members were treated with clear declaration of conflict of interest and decided by the PC chair without conflict of interest. In the end, based on the reviews, the PC had decided to accept two full papers, one work-in-progress paper, and two position papers. Five paper presentations are part of the SEA4DQ 2022's program together with two keynotes, one project presentation (InterQ), and a panel discussion.

SEA4DQ 2022 features Prof. Dr. Andreas Metzger, Head of Adaptive Systems and Big Data Applications from University of Duisburg-Essen, Germany, and Prof. Dr. Foutse Khomh, Head of SoftWare Analytics and Technologies (SWAT) Lab, University of Montréal, Canada as two keynote speakers. The first keynote speaker Andreas Metzger addresses *Data Quality Issues in Online Reinforcement Learning for Self-adaptive Systems*. The second keynote speaker Foutse Khomh addresses *Quality and Model Under-Specification Issues*. Furthermore, the accepted contributions, selected carefully by the program committee, show the research trends to address data quality issues are being intensified. Jørgen Stang, Dirk Walther, Per

Myrseth report their full paper on *Data Quality as a Microservice - an ontology and rule based approach for quality assurance of sensor data in manufacturing machines*. Muhammad Azmi Umer, Aditya Mathur and Muhammad Taha Jilani present their full paper *Effect of Time Patterns in Mining Process Invariants for Industrial Control Systems: An Experimental Study.* Valentina Golendukhina, Harald Foidl, Michael Felderer and Rudolf Ramler provide in their work-in-progress paper *Preliminary Findings on the Occurrence and Causes of Data Smells in a Real-World Business Travel Data Processing Pipeline*. Maryna Waszak, Terje Moen, Sølve Eidnes, Alexander Stasik, Anders Hansen, Gregory Bouquet, Antoine Pultier, Xiang Ma, Idar Tørlen, Bjørn Rune Henriksen, Arianeh Aamodt, Dumitru Roman address *Data Quality Issues for Vibration Sensors: A Case Study in Ferrosilicon Production* in their position paper. And last but not least, Dumitru Roman, Antoine Pultier, Xiang Ma, Ahmet Soylu, Alexander G.Ulyashin present *Data Quality Issues in Solar Panels Installations: A Case Study* in their position paper.

We would like to thank the program committee members and all reviewers for their work in evaluating the submissions. We also thank the SEA4DQ 2022 organizers for their assistance in the preparation of the workshop and the editors of ESEC / FSE 2022 for help in publishing these proceedings.

Singapore
November 2022

Phu H. Nguyen
Maria Chiara Magnanini
Sagar Sen

SEA4DQ 2022 chairs

SEA4DQ 2022 Organization

Organizing Committee

General Chair
Phu H. Nguyen — SINTEF, Norway

Program Co-Chairs
Maria Chiara Magnanini — Politecnico di Milano, Italy
Sagar Sen — SINTEF, Norway

Moderator, Co-Web Chairs
Beatriz Bretones Cassoli — TU Darmstadt, Germany
Nicolas Jourdan — TU Darmstadt, Germany

Publicity Chair
Mikel Armendia — Tekniker, Spain

Program Committee

Abhilash Anand, DNV AS, Norway
Enrique Garcia Ceja, Optimeering, Oslo, Norway
Sudipto Ghosh, Colorado State University, USA
Helena Holmström Olsson, Malmö University, Sweden
Frank Alexander Kraemer, NTNU, Norway
Felix Mannhardt, KIT-AR, Germany
Dusica Marijan, Simula Research Laboratory, Norway
Andreas Metzger, University of Duisburg-Essen, Germany
Jan Nygård, Cancer Registry of Norway, Norway
Karl John Pedersen, DNV AS, Norway
Dimitra Politaki, INLECOM, Greece
Dumitru Roman, SINTEF / University of Oslo, Norway
Marc Roper, University of Strathclyde, UK
Helge Spieker, Simula Research Laboratory, Norway
Jean-Yves Tigli, Université Côte d'Azur, France
Hong-Linh Truong, Aalto University, Finland
Katinka Wolter, Free University of Berlin, Germany
Amina Ziegenbein, Technische Universität Darmstadt, Germany

SEA4DQ 2022 Sponsors

The SEA4DQ 2022 Workshop is sponsored by the research projects InterQ and DAT4.Zero that are funded by the European Union's Horizon 2020 Research and Innovation programme under the grant agreement numbers 958357 (InterQ) and 958363 (DAT4.Zero).

https://interq-project.eu/

https://dat4zero.eu/

https://ec.europa.eu/

Contents

Frontmatter

Keynotes

Software Engineering and AI for Data Quality

Data Quality Issues in Online Reinforcement Learning for Self-Adaptive Systems (Keynote)

Andreas Metzger
andreas.metzger@paluno.uni-due.de
paluno (The Ruhr Institute for Software Technology), University of Duisburg-Essen
Essen, Germany

ABSTRACT

Online reinforcement learning is an emerging machine learning approach that addresses the challenge of design-time uncertainty faced when building self-adaptive systems. Online reinforcement learning means that the self-adaptive system can learn from data only available at run time. After introducing the fundamentals of self-adaptive systems and reinforcement learning, the keynote discusses three relevant issues and recent solutions related to data quality in online reinforcement learning for self-adaptive systems.

CCS CONCEPTS

• **Software and its engineering → Designing software**.

KEYWORDS

Machine learning, adaptive system

ACM Reference Format:
Andreas Metzger. 2022. Data Quality Issues in Online Reinforcement Learning for Self-Adaptive Systems (Keynote). In *Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (SEA4DQ '22), November 17, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3549037.3570194

1 INTRODUCTION

A self-adaptive system can modify its structure and behavior at run time based on its perception of its environment, itself, and its requirements. Via self-adaptation the system can maintain its requirements in the presence of dynamic environment changes [4]. Examples of self-adaptive systems include elastic cloud systems, intelligent IoT systems, and proactive process management systems.

When developing a self-adaptive system, developers face the challenge of design-time uncertainty. They have to anticipate potential environment states and the precise effect of an adaptation in a given environment state. However, oftentimes the knowledge available at design time is not sufficient to do so [5].

This keynote explores the opportunities but also challenges that modern machine learning algorithms offer in engineering self-adaptive systems in the presence of design-time uncertainty. It will focus on online reinforcement learning as an emerging approach. Online reinforcement learning enables the self-adaptive system to learn from data only available at run time.

The keynote addresses the following issues and recent solutions related to data quality in online reinforcement learning for self-adaptive systems: (*a*) data drift [3], (*b*) data sparsity [2], and (*c*) data non-transparency [1]. The keynote also provides a critical discussion and an outlook on future research opportunities.

2 BIOGRAPHY

Prof. Dr. Andreas Metzger is an adjunct professor at the University of Duisburg-Essen and heads the "Adaptive Systems" group at paluno, the Ruhr Institute for Software Technology. His current research interests include the use of machine learning in software engineering and business process management. He is the steering committee vice chair of the Networked European Software and Services Initiative and was deputy general secretary of the European Big Data Value Association from 2015 to 2021. Among other leadership roles, he was the technical coordinator of the European lighthouse project TransformingTransport.

ACKNOWLEDGMENTS

Research leading to these results received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 871493: DataPorts.

REFERENCES

[1] Felix Feit, Andreas Metzger, and Klaus Pohl. 2022. Explaining Online Reinforcement Learning Decisions of Self-Adaptive Systems. In *IEEE International Conference on Autonomic Computing and Self-Organizing Systems, ACSOS 2022, Virtual, September 19-23, 2022*, Elisabetta Di Nitto, Ilias Gerostathopoulos, Kirstie Bellman, and Sven Tomforde (Eds.). IEEE, 51–60.

[2] Andreas Metzger, Clément Quinton, Zoltán Ádám Mann, Luciano Baresi, and Klaus Pohl. 2022. Realizing Self-Adaptive Systems via Online Reinforcement Learning and Feature-Model-guided Exploration. *Computing* (2022), 1–22.

[3] Alexander Palm, Andreas Metzger, and Klaus Pohl. 2020. Online Reinforcement Learning for Self-adaptive Information Systems. In *32nd International Conference on Advanced Information Systems Engineering, CAiSE 2020, Grenoble, France, June 8-12, 2020 (LNCS, Vol. 12127)*, Schahram Dustdar, Eric Yu, Camille Salinesi, Dominique Rieu, and Vik Pant (Eds.). Springer, 169–184.

[4] Danny Weyns. 2020. *An Introduction to Self-adaptive Systems: A Contemporary Software Engineering Perspective.* John Wiley & Sons.

[5] Danny Weyns, Ilias Gerostathopoulos, Nadeem Abbas, Jesper Andersson, Stefan Biffl, Premek Brada, Tomas Bures, Amleto Di Salle, Patricia Lago, Angelika Musil, Juergen Musil, and Patrizio Pelliccione. 2022. Preliminary Results of a Survey on the Use of Self-Adaptation in Industry. In *17th Intl Symp. on Software Engineering for Adaptive and Self-Managing Systems, SEAMS@ICSE 2022.* ACM/IEEE, 70–76.

SEA4DQ '22, November 17, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9459-8/22/11.
https://doi.org/10.1145/3549037.3570194

Data Quality and Model Under-Specification Issues (Keynote)

Foutse Khomh
Polytechnique Montréal
Canada
foutse.khomh@polymtl.ca

ABSTRACT

Nowadays, we are witnessing an increasing demand in both industry and academia for exploiting Deep Learning (DL) to solve complex real-world problems. However, the performance of these high-capacity learners is currently bounded by the quality and volume of their underlying training data. The use of incomplete, erroneous, or inappropriate training data, and the implementation of poor data management practices in a training pipeline often result into unreliable, biased, or under specified models. In this talk, I will report about some recent research works that we have conducted to identify best practices of data management for DL. I will also report about recent techniques and tools that we have developed to help detect the root cause of model under-specification issues early on during a DL training process.

ACM Reference Format:
Foutse Khomh. 2022. Data Quality and Model Under-Specification Issues (Keynote). In *Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (SEA4DQ '22), November 17, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 1 page. https://doi.org/10.1145/3549037.3570195

BIOGRAPHY

Foutse Khomh is a Full Professor of Software Engineering at Polytechnique Montréal, Canada CIFAR AI Chair on Trustworthy Machine Learning Software Systems, and FRQ-IVADO Research Chair on Software Quality Assurance for Machine Learning Applications. He received a Ph.D. in Software Engineering from the University of Montreal in 2011, with the Award of Excellence. He also received a CS-Can/Info-Can Outstanding Young Computer Science Researcher Prize for 2019. His research interests include software maintenance and evolution, machine learning systems engineering, cloud engineering, and dependable and trustworthy ML/AI. His work has received four ten-year Most Influential Paper (MIP) Awards, and six Best/Distinguished Paper Awards. He also served on the steering committee of SANER (chair), MSR, PROMISE, ICPC (chair), and ICSME (vice-chair). He initiated and co-organized the Software Engineering for Machine Learning Applications (SEMLA) symposium and the RELENG (Release Engineering) workshop series. He is co-founder of the NSERC CREATE SE4AI: A Training Program on the Development, Deployment, and Servicing of Artificial Intelligence-based Software Systems, and one of the Principal Investigators of the DEpendable Explainable Learning (DEEL) project. He is on the editorial board of multiple international software engineering journals and is a Senior Member of IEEE.

SEA4DQ '22, November 17, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9459-8/22/11.
https://doi.org/10.1145/3549037.3570195

Data Quality as a Microservice: An Ontology and Rule Based Approach for Quality Assurance of Sensor Data in Manufacturing Machines

Jørgen Stang*
Dirk Walther*
Per Myrseth*
jorgen.stang@dnv.com
dirk.walther@dnv.com
per.myrseth@dnv.com
DNV
Oslo, Norway

ABSTRACT

The manufacturing industry is continuously looking for production improvements resulting in high quality production, reduced waste and competitive advantages. In this article, ontologies, semantic rule logic and microservices have been deployed to suggest a system for quality assurance of manufacturing machine data. The existing upper ontology for manufacturing service description has been used to define both the physical assets as well as the data quality requirements. The system is used to both operationalize data quality monitoring by semantic technology as well as enabling up-front modelling of data quality requirements. The approach is illustrated by a specific speed-feed case for manufacturing machines but could easily be extended to other manufacturing use-cases or even to other industries.

CCS CONCEPTS

• **Computer systems organization** → **Architectures**; • **Theory of computation** → *Semantics and reasoning*; • **Applied computing** → Industry and manufacturing.

KEYWORDS

Data Quality, Manufacturing Machines, Sensor Data, Microservices, Ontolologies, IoT

ACM Reference Format:
Jørgen Stang, Dirk Walther, and Per Myrseth. 2022. Data Quality as a Microservice: An Ontology and Rule Based Approach for Quality Assurance of Sensor Data in Manufacturing Machines. In *Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (SEA4DQ '22), November 17, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3549037.3561272

*All authors contributed equally to this research.

SEA4DQ '22, November 17, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9459-8/22/11…$15.00
https://doi.org/10.1145/3549037.3561272

1 INTRODUCTION

Internet of Things (IoT) and Industry 4.0 have already enabled digital transformation of knowledge, processes and services in most industries. Advanced technology is both common-place and off-the-shelf, providing means for quality improvements and efficiency gains. In the manufacturing industry, machines and devices are equipped with sensors to monitor operational characteristics such as speed, heat, vibrations and more. These characteristics are subsequently used to detect deviations, failure modes, inconsistencies and other events adversely affecting production quality. Traditionally, this monitoring would rely on humans to detect by noise, visuals, touch or even smell. Considering that the cost of rework and waste is significant [24] the digital capabilities providing continuous monitoring and analysis of operations is a game changer. In this new reality, the quality of the sensor data used for monitoring and issue handling will be a critical success factor. If left unattended, sensors, just as the physical assets, will malfunction, drift, freeze, misalign or plainly break. Subsequently, this will yield wrong information which in turn can both abate failure detection as well as trigger costly uncalled-for operations. This paper outlines how ontologies based on the Manufacturing Service Description Language (MSDL) can be used with data quality requirements expressed by the Semantic Web Rule Language (SWRL) as model constraints. The specific data quality rules are implemented as a microservice that can be reused and deployed both across manufacturing machines as well as across domains.

The work described here is intended to contribute towards the ambition of zero defect manufacturing [13], where all process, product and data is monitored to ensure any deviations are captured and handled before production quality is affected.

2 OVERALL SYSTEM ARCHITECTURE

The described system uses a 3 layered architecture and each component is briefly described in the next sections. The main objective is to provide data quality monitoring by automatic reasoning (inference) for the manufacturing machine signals. Figure 1 shows the 3 layers; (1) model definition by ontology on top, (2) semantics rule logic in the middle and (3) the data quality service at the bottom. All three layers are required to efficiently represent:

- Ontology – Model constraints (e.g. machine has a feed);

- Rule – Semantic constraints (e.g. correlation coefficient should be more than 0.8 between feed and cutting speed for a single axis machines); and
- Metric – Data Quality as a Service (e.g. calculate correlation coefficient).

The following is a brief description of Machine Signal, Ontology, Rule, Metric and Result as shown in Figure 1.

Machine signals are received by event-hubs, streaming APIs, historical data loads or other integration methods. Usually signals are buffered to enable data quality metrics to be performed on more than a single datapoint. Even though some metrics are useful on single data points (say range and code list validation), most metrics will require a dataset as input. Also, performance issues will often debate operations on single datapoints.

The buffered dataset is uploaded to the ontology and validated according to the defined relations. This utilises the capabilities of the ontology to define complex models and assign a meaning to the relations itself. The particular ontology used here is described in a later section. The valid model can subsequently be queried by rules to determine compliance with requirements. Ontologies have reasoning capabilities and in addition we deploy a dedicated rule definition language such as SWRL [12], SHACL [16] or SPIN [15]. Rule languages will add query capabilities that is not provided directly by the ontology itself, however, the model semantics are utilised to provide "smart" rule execution, meaning, depending on the machine configuration, different rules and different rule parameters will be used.

The rule will in turn trigger a data quality metric. The metric could be compute intensive and rely on complex algorithms and is therefore implemented in compiled code that executes on scalable compute resources such as Kubernetes clusters [10], virtual or on-premise machines. The choice of compute resource will often rely on security issues and enterprise strategy for cloud or federation.

The data quality assessment result is subsequently used for alarms, notifications and trend analysis, and will ultimately drive the improvement processes to ensure that the data produced by the manufacturing machine sensors are fit for use and that operations are performed within acceptable risk.

The next sections will describe the 3 main components; Ontology, Rule and Metric in more detail.

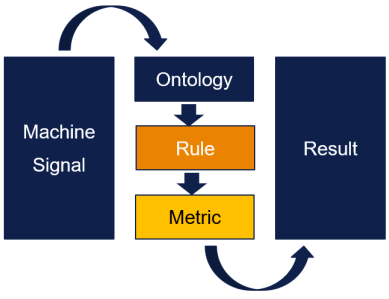


Figure 1: System Architecture

3 ONTOLOGY FOR MANUFACTURING

Ontologies are generally implemented to model complex relations in information models. Properly defined, ontologies can be used to both represent knowledge in the model as well as to provide inference capabilities (automatic reasoning) enabling advanced queries. Knowledge is represented by semantics that define the meaning of a relation, whereas inference is used to query the model by semantics. These capabilities makes ontologies a very powerful modelling and validation tool [11]. As a very simple example, in a manufacturing machine, an ontology can represent a carving machine with 1 tool which in turn can have spindle positions that move in 1-3 directions (x,y,z). The ontology can hence validate that the machine configuration is correct (1 tool and 1-3 directions), in addition, the inference mechanism can use the number of directions present to infer other characteristics. This example is elaborated further in a later section.

Well defined ontologies are a prerequisite to ensure scaling on performance and model complexity. Ontologies are commonly divided into layers which represent different levels of abstraction [7]. The top ontology, or upper ontology, will focus on abstract entities and the middle layer adds domain specific entities. Based on the upper and middle ontology, specialised entities and constructs can be added to provide custom implementations. This provides for an extensible and flexible modelling paradigm, at the same time, overlapping lower ontologies in any layer should be avoided. To that end, several industries have established fora for the development and exchange of common ontologies [14] [19] [23] [20]. The Manufacturing Service Description Language (MSDL) was first introduced in 2006 [2] and further revised in 2019 [5]. It was developed to support interoperability and advanced reasoning for manufacturing services within a Virtual Enterprise (VE) [2]. Except for mentions in recent PhD theses [18] [17] and several articles [5] [3], there is no evidence of extensive adoption of MSDL in the industry [4], however, currently, MSDL has been incorporated into the Industrial Ontologies Foundry (IOF) initiative and will probably as a consequence gain more traction. In this work we use MSDL as the upper ontology as the current version of MSDL is well suited for our purpose. By nature, ontologies evolve and are designed to be extended for special purposes. New versions of MSDL or other suitable upper ontologies for manufacturing could hence also be incorporated in the future.

As shown in Figure 2, MSDL divides the manufacturing service into supplier, process and resource. Machining capabilities is a resource which again includes physical components such as axes, tables, spindle and tool [28]. We also include the entities feed and clock as the relationship between cutting speed (spindle) and feed rate is an important driver for product quality [28] and, also, clock speed and synchronization is required for both analysis and monitoring. Figure 2 show the MSDL based ontology used in the work presented here.

4 ONTOLOGY RULE LOGIC EXPRESSIONS

Several formal constructs have been implemented to add rule logic capabilities to ontologies [12] [16] [15]. This is useful to express specific rules that is not supported by the ontology itself. The ontology described in the previous chapter and shown in Figure 2 can

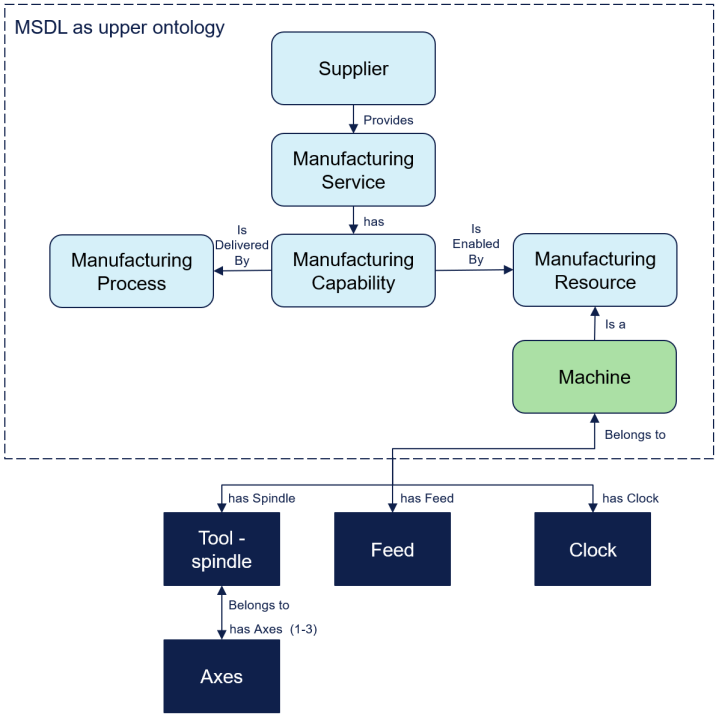


Figure 2: Ontology for manufacturing (MSDL) used as upper ontology

express that a machine has 1 feed, 1 spindle with 1-3 axes and 1 clock. However, rule logic expressions can be used to determine relationships between specific machine configurations and required query results. In the context of data quality, there could be different requirements to correlation coefficient between spindle speed and feed rate depending on the number of axes. For 1 axis, a typical drilling operation, the correlation could be linear, whereas for spatial operations with 3 axes the correlation could be weaker or even non-existent. Hence, rule logic can be expressed as

> *SELECT* rule *WHERE* machine hasFeed *COUNT* 1 *AND* hasSpindle *COUNT* 1 *AND* has Axes *COUNT* 3 *AND* result < 0.4

will return the appropriate data quality rule. The data quality rule will be executed by external service (DQaaS) and the result yields acceptance criteria for this particular data quality metric. The above rule illustrates how ontologies and rules can be layered to provide powerful querying capabilities.

Rule languages are currently more immature than ontologies and hence not standardised and formalized to the same extent. The above statement is only for illustration and do not adhere to a specific rule format. The specific number used as rule result threshold (0.4) is explained later.

5 DATA QUALITY AS A MICROSERVICE

The term data quality as a service (DQaaS) denotes an existing library of data quality metrics that can be accessed as a cloud service or it can be deployed to local clusters [6] [13]. The label 'micro' simply indicates the service is stateless, specialised and will require some level of orchestration by an API-gateway or client applications. The DQaaS API provide access to methods as endpoints and the CPU can be provided by Kubernetes clusters, virtual machines or on-premise servers. The service provides the bottom layer of the architecture where compute intensive operations are performed. The service is implemented in Python using Pandas, Numpy, Great Expectations and other standard modules. The API complies to OpenAPI 3.0 [25] and is implemented with FastAPI [27]. The data quality metrics are predominantly geared towards IoT time series data in the format *timestamp — signal — value* which easily adopts itself to manufacturing machine signals. Some example metrics are shown in Figure 3. Duplicates, missing values, invalid values, invalid distributions and other anomalies are covered by the service, currently there are approximately 20 rules available. The rules are intended to monitor sensors for anomalies such as miscalibration, drift, freeze, downtime, clock-synchronization, noise, malfunction and others. Some metrics are defined in more detail in a later section. The following data quality issues are shown in Figure 3: Time collision (duplicate timestamps for same signal), outside range (where range is defined from min to max), Rate of Change (RoC), missing data (values or records) and drift.

Data quality is often defined by data quality frameworks [22] [8] and there are also dedicated standards such as ISO 8000 [1]. Typically the frameworks will categorize and suggest specific metrics definitions. ISO 8000 offers the distinction between syntax, semantic and pragmatic data quality, meaning format errors (wrong data

type), invalid data (according to real world asset) and use case dependent respectively. The DQaaS is predominantly concerned with the semantic category and the metrics are based on common issues typically encountered for time series data [21] [9]. The pragmatic, or use case dependent category, is often used for different system configurations, for example, use cases involving analytics (prediction or trending) will require data quality measured at a high resolution. On the other hand, detecting failure modes with long *p–f* intervals (time from detection of *potential failure* to *–failure* happens) could require measurements at lower resolutions. Hence, it should be noted that data that have good quality for one use-case (say long p–f intervals) could be unfit for other use cases (say predictive analytics).

In addition to the broad categories syntactic, semantic and pragmatic, ISO 8000 also offers more detailed data quality characteristics and data quality anomalies. In the use case presented in a later section, we look at the characteristic called *consistency* (between feed rate and spindle speed) and the resulting anomaly *drift*. The consistency is calculated as the correlation coefficient for the related sensor signals. Drift occurs when related sensor signals experience increasing deviations, this is also shown graphically in Figure 3.

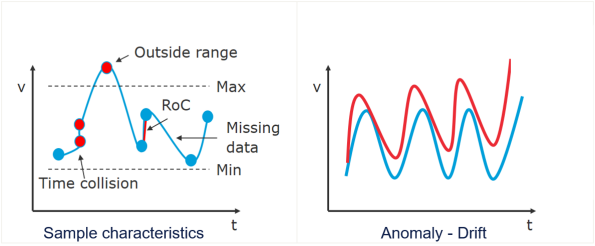


Figure 3: Sample data quality metrics for time series data

6 CONCEPTUAL MODEL

The overall architecture shown previously can be expanded to a more detailed conceptual model. The MSDL upper ontology is extended to include entities that will serve as input to the data quality service. The rule semantics connects related entities and the relevant rules defined in the data quality service. In Figure 4 the manufacturing machine is modelled by asset components, however there is no classification or modelling of the data quality rules. This asset-in-focus approach is commonly used to support digital twins and data interoperability, meaning there should be a consensus on how to represent the physical asset digitally. When this consensus is reached, the same unified model can be reused across machines as well as enterprises. The data quality metrics will then be called based on asset configuration and data quality requirements defined in the model. If a machine has a clock it should have a frequency and the frequency should be according to a given value. If the requirement is 1 Hz frequency and the monitored frequency is higher, this should trigger a notification that defines an action to be taken to mitigate the issue.

Alternatively, the ontology could emphasize on the data quality rules rather than the asset, focusing on data integration rather than interoperability. Any proprietary model could connect to the rule ontology by mapping individual terms to a rule vocabulary. This alleviates the requirements for a common information model and relies on the rule classification to define correct semantics. This approach has not been pursued further here but could prove useful for a use cases where there are complex requirements or regulations that should be applied to disparate data sources. The regulations (say GDPR) can then be modelled by the means described here and subsequently be applied to proprietary systems (say CRM).

The MSDL based ontology has attributes that describes physical features such as clock frequencies, feed rates and spindle positions. These attributes are managed by the query mechanism to trigger data quality rules. The data quality rules shown here are not exhaustive but represents commonly used metrics:

- Noise – Measures deviation between values of same attribute with a sideways shift, random deviations indicates noise in signal
- Frequency – Calculates lag between sorted timestamps and compares to requirement
- Duplicates – Identical timestamp for same signal
- Range/Rate of change – According to defined min-max values / according to allowed rate of change
- Deviation – Allowed difference between data points
- Distributions – Statistical distribution requirements such as normal, chi-square, Smirnov and others
- Correlations – Calculates correlation coefficient for related attribute series

7 USE CASE

The implemented use case is shown in the below figure. The main motivation for looking at the spindle and the feed-rate is the effect any mis-configuration of these parameters will have on the end product. Surface and finishing quality will deteriorate significantly if the material is fed out-of-sync with the cutting speed, tools can be damaged and material is wasted [28]. Therefore, careful monitoring of these critical parameters is required, and, subsequently the data quality should also be monitored. As mentioned in a previous section we will focus on the data quality characteristic *consistency* and the data quality anomaly *drift* as defined by ISO 8000. Consistency is calculated as the correlation coefficient for the two related sensors, feed rate and spindle speed. The relationship between feed rate and spindle speed will depend on material type, cutting axes and others [26]. The below formulae defines the mathematical relationships between cutting speed, spindle speed and feed [28]:

$$\text{Cutting speed: } V_c = \frac{\pi \; x \; D \; x \; n}{1000}$$

$$\text{Spindle speed: } n = V_c - \pi - D \; x \; 1000$$

$$\text{Feed: } V_f = n \; x \; f_z \; x \; Z$$

where D is the spindle diameter, f_z is feed per tooth and Z is number of flutes or teeth.

In this case it suffices to state that for any given machine configuration the relationship is constant. In addition, the ontology will yield model cardinality, as an example, the feed-speed correlation

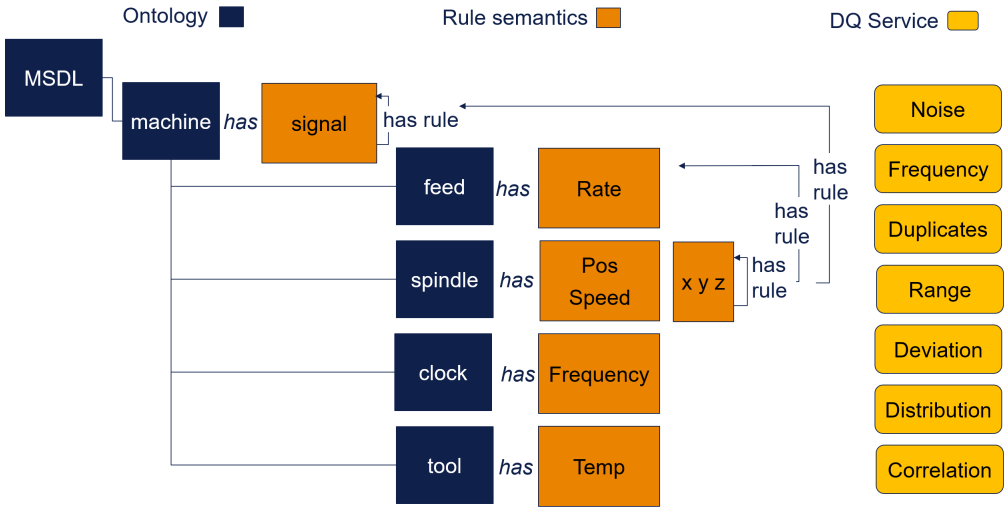


Figure 4: Conceptual model

depends on the number of axes and hence the automatic reasoning capabilities will handle correlation rules for different spatial carving capabilities (drilling, surface, moulding). Figure 5 show the speed-feed correlation for a moulding machine with spatial spindle positions (x,y and z) during two time intervals, t1 and t2. As shown, the feed will have different correlation to the 3 axes during the moulding process. The spindle z-position has high correlation in the first period t1 (0.4 from 07:38:30 to 07:39:00) whereas both z and x position has high values for the last period t2 (0.3 and 0.36 respectively from 07:40:00 to 07:41:00). If we assume the data used for the analysis have good quality, this can be used to set the threshold to detect any sensor data quality issues for the relevant sensors. The actual threshold value can be set up in a number of ways, here we simply say the squared value for the correlation coefficient for all axes should be above 0.4 (measured values are 0.42 and 0.48 for the two periods shown in the figure). This could be an oversimplification, also, we do not know if the data has already drifted or if there are any other data quality issues in the sample data, but it will suffice as an illustration for this use case.

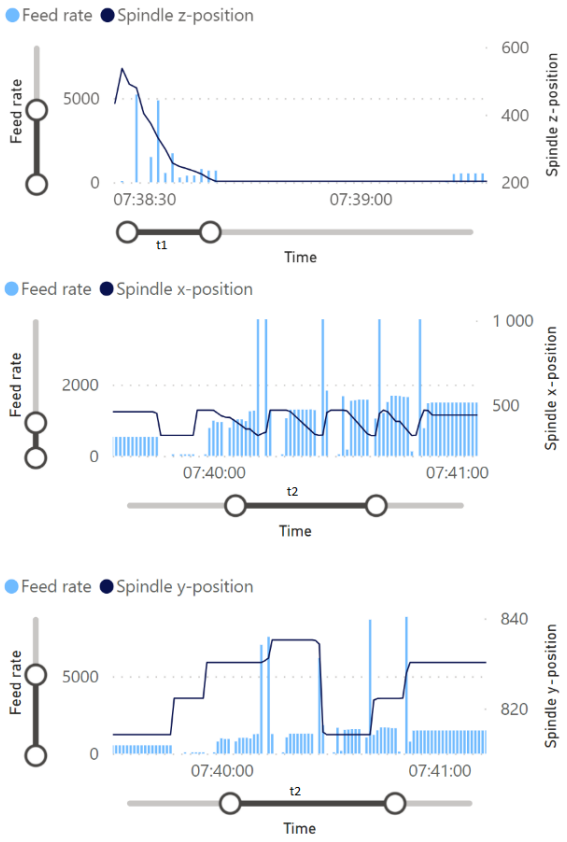
In addition to speed-feed correlation, the use case also includes rules for machine counter, status, clock frequency and valid ranges. The following is a list of the relevant rules, label in italic refers to Figure 6:

- *api:rule corr > 0.4* — feed/speed correlation coefficient should be above this value
- *api:rule <min ,max>* — Range for valid values, between min and max
- *api:rule <0,inf>* — Values should be above 0
- *api:rule f=1 Hz* — Timestamp should have this frequency
- *api:rule >0 then ON* — If speed is above zero then machine status should be ON
- *api:rule [ON,OFF]* — Machine status should be ON or OFF
- *api:rule n+1>n* — Machine running counter should always be increasing

The purpose of the data quality rules is to detect anomalies in the data and this should trigger a root cause analysis to define activities to support continuous improvement. The above list is not exhaustive and domain expertise should be used to define additional rules that can be added to the service and again triggered by the rules and constraints defined in the model. The use case also illustrates how the knowledge model (ontology) can be used in an operational setting with constraints and requirements. Data from manufacturing machines can be loaded into the ontology in real time and described mechanisms will continuously evaluate compliance to requirements. Also, this modelling approach ensures that data quality requirements are considered up front as part of modelling and design, and not as an ad hoc afterthought, which is often the case.

8 CONCLUSION

The Manufacturing Service Description Language (MSDL) have been used as a basis for a tentative extension to express data quality requirements for a simple manufacturing machine with clock, feed and spindle. A generic data quality service for sensor data can be used to calculate the data quality requirements based on semantic rule expressions such as SWRL. The data quality service was implemented in Python based on Great Expectations and deployed as a microservice on a cloud platform. The work described here shows how ontologies can be used to both model the knowledge (terminology/structure/semantics) of the asset as well as defining requirements and constraints to the production process itself. The resulting regime provides quality assurance of complex assets, even digital twins, and will apply relevant data quality rules based on asset configurations. One example was used for illustration where the ontology will distinguish between linear carving (drilling) and spatial carving (moulding) and apply appropriate requirement to the correlation between machine status, clock frequency, spindle speed and feed rate.



Time t1	*Axis*	*Feed rate corr*
07:38:30 - 07:39:00	Spindle z-position	0.4
07:38:30 - 07:39:00	Spindle y-position	0.086
07:38:30 - 07:39:00	Spindle x-position	0.09
	Total (squared)	0.42

Time t2	*Axis*	*Feed rate corr*
07:40:00 - 07:41:00	Spindle z-position	0.3
07:40:00 - 07:41:00	Spindle y-position	0.14
07:40:00 - 07:41:00	Spindle x-position	0.36
	Total (squared)	0.48

Figure 5: Correlation values for feed rate and spindle positions for sample data for a moulding machine

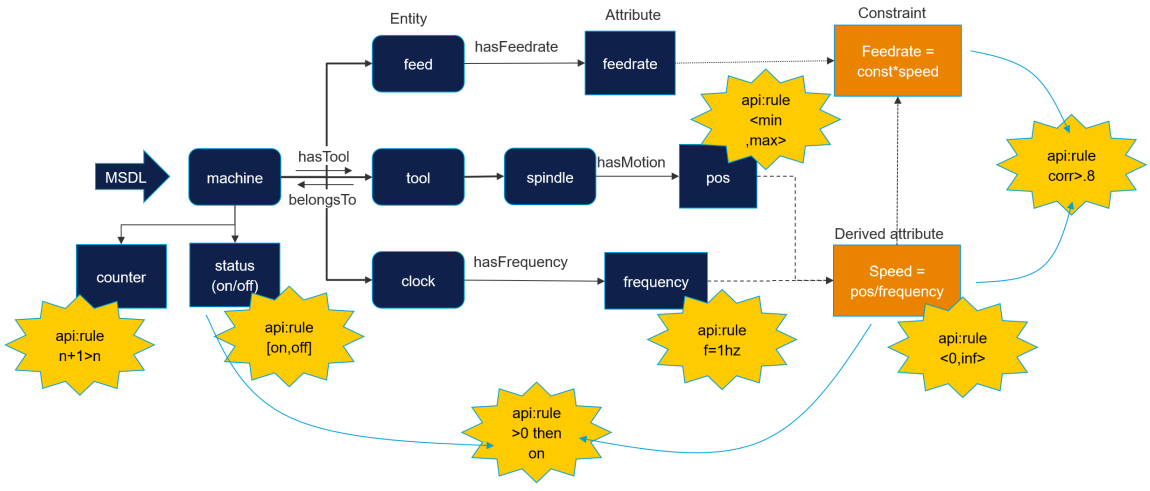


Figure 6: Use case

Jointly, the ontologies and the microservices can support both traditional model verification as well as process verification. This combination provides a manner to include and design for data quality during early design phases. Traditionally data quality tends to be left as an after-thought and implemented in an ad-hoc manner. Considering the significant importance of data quality in digital

processes, the data quality requirements should be defined and implemented up-front.

Further work should look in to how the data quality result can be represented as an ontology itself and subsequently used as a driver for improvement activities and risk analysis. The ontology described here should be further expanded and formalised to support more advanced use-cases, also, the data quality service can be extended with additional rules. Ontologies will scale on both complexity and size, also, the microservice and cloud based architecture will deploy to any given compute capabilities.

ACKNOWLEDGEMENTS

This project has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 958357 and it is an initiative of the Factories-of-the-Future (FoF) Public Private Partnership.

REFERENCES

[1] ISO 8000-8. 2015. *Data quality — Part 8: Information and data quality: Concepts and measuring.* Retrieved June 23, 2022 from https://www.iso.org/standard/60805.html

[2] Farhad Ameri and Debasish Dutta. 2006. An upper ontology for manufacturing service description. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Vol. 42578. 651–661.

[3] Farhad Ameri, Christian McArthur, Bahram Asiabanpour, and Mohammad Hayasi. 2011. A Web-based Framework for Semantic Supplier Discovery for Discrete Part Manufacturing. In *Proceedings of NAMRI/SME*, Vol. 39.

[4] Farhad Ameri, Dusan Sormaz, Foivos Psarommatis, and Dimitris Kiritsis. 2022. Industrial ontologies for interoperability in agile and resilient manufacturing. In *Industrial Journal of Production Research*, Vol. 60.

[5] Damiano Arena, Farhad Ameri, and Dimitris Kiritsis. 2019. Skill Modelling for Digital Factories. In *IFIP International Conference on Advances in Production Management Systems (APMS)*. 318–326.

[6] Veracity by DNV. 2022. *Veracity for Developers.* Retrieved June 23, 2022 from https://www.veracity.com/

[7] Oscar Cabrera, Xavier Franch, and Jordi Marco. 2015. A Middle-Level Ontology for Context Modelling. In *International Conference on Conceptual Modeling.*

[8] Corinna Cichy and Stefan Rass. 2019. An Overview of Data Quality Frameworks. *IEEE Access* 7 (2019), 24634–24648. https://doi.org/10.1109/ACCESS.2019.2899751

[9] DNV. 2017. Data Quality Assessment for Sensor Systems and Time-Series Data. In *DNV Report No. 2017-0058*, Vol. 1.01.

[10] Red Hat. 2020. *What is a Kubernetes cluster?* Retrieved June 23, 2022 from https://www.redhat.com/en/topics/containers/what-is-a-kubernetes-cluster

[11] Claudia Hess and Christop Schlieder. 2006. Ontology-based verification of core model conformity in conceptual modeling. In *Computers, Environment and Urban Systems*, Vol. 30.

[12] Ian Horrocks, Peter F. Patel-Schneider, Harold Boley, Said Tabet, Benjamin Grosof, and Mike Dean. 2004. *SWRL: A Semantic Web Rule Language Combining OWL and RuleML.* Retrieved June 23, 2022 from https://www.w3.org/Submission/SWRL/

[13] InterQ. 2022. *Project for zero defect manufacturing.* Retrieved June 23, 2022 from https://interq-project.eu/

[14] IOF. 2022. *Industrial Ontologies Foundry (IOF) website.* Retrieved June 23, 2022 from https://www.industrialontologies.org/

[15] Holger Knublauch, James A. Hendler, and Kingsley Idehen. 2011. *SPIN - Overview and Motivation.* Retrieved June 23, 2022 from https://www.w3.org/Submission/spin-overview/

[16] Holger Knublauch and Dimitris Kontokostas. 2017. *Shapes Constraint Language.* Retrieved June 23, 2022 from https://www.w3.org/TR/shacl/

[17] Roman Korecky. 2022. *Ontologies in Industry 4.0: Standards, Applications, and Methodologies.* Master's thesis. Department of Information Systems and Operations, Vienna University of Economics and Business.

[18] Yongxin Liao. 2013. *Semantic Annotations for System Interoperability in a PLM Environment.* Retrieved June 23, 2022 from https://tel.archives-ouvertes.fr/tel-00904822/file/20131114_Thesis_Defence_YongxinLIAO_Version_1.7.pdf

[19] OAGI. 2022. *Open Application Group (OAGi) website.* Retrieved June 23, 2022 from https://oagi.org/

[20] POSC-CAESAR. 2022. *POSC Caesar Association.* Retrieved June 23, 2022 from https://www.posccaesar.org/

[21] DNV Recommended Practice. 2021. *Assurance of sensor systems.* Retrieved June 23, 2022 from https://rules.dnv.com/servicedocuments/dnv/#!/industry

[22] Roseanne Price and Graeme Shanks. 2005. A Semiotic Information Quality Framework: Development and Comparative Analysis. In *Journal of Information Technology*, Vol. 20.

[23] READI. 2022. *Requirement Asset Digital Lifecycle Information (READI) website.* Retrieved June 23, 2022 from https://readi-jip.org/

[24] Anna-Katrina Shedletsky. 2019. *Manufacturing Wastes 10% Of The GWP Every Year. Here's Why.* Retrieved June 23, 2022 from https://www.forbes.com/sites/annashedletsky/2019/10/18/manufacturing-wastes-10-of-the-global-gdp-every-year-heres-why/?sh=5c5b80351098

[25] Swagger. 2022. *OpenAPI Specification.* Retrieved June 23, 2022 from https://swagger.io/specification/

[26] Christopher Tate. 2014. *Understanding cutting equations.* Retrieved June 23, 2022 from https://www.ctemag.com/news/articles/understanding-cutting-equations

[27] Tiangolo. 2022. *FastAPI framework, high performance, easy to learn, fast to code, ready for production.* Retrieved June 23, 2022 from https://fastapi.tiangolo.com/

[28] WAYKEN. 2022. *Feed Rate And Cutting Speed: What's The Difference In CNC Machining.* Retrieved June 23, 2022 from https://waykenrm.com/blogs/feed-rate-and-cutting-speed-difference-in-cnc-machining/

Effect of Time Patterns in Mining Process Invariants for Industrial Control Systems: An Experimental Study

Muhammad Azmi Umer
CodeX LLC, Karachi
Karachi Institute of Economics and Technology
Pakistan
mazmi@codexnow.com

Aditya Mathur
Singapore University of Technology and Design
Singapore
aditya_mathur@sutd.edu.sg

Muhammad Taha Jilani
Karachi Institute of Economics and Technology
Pakistan
m.taha@kiet.edu.pk

ABSTRACT

Machine Learning is playing a crucial role in the design of intrusion detectors for Industrial Control Systems (ICS). Intrusion Detection Systems (IDS) rely on data obtained from an operational ICS. Such datasets contain multiple time series, one for each process variable. In this work, we explore how such time series can be exploited to understand the effect of time patterns in mining the process invariants, i.e., conditions on process state variables. We use the knowledge gained through the time patterns to determine the optimal data collection size for generating the invariants. The study reported here was conducted using the operational data obtained from a water treatment plant.

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**; • **Computing methodologies → Anomaly detection**; **Rule learning**; • **Information systems → Association rules**.

KEYWORDS

Machine Learning, Intrusion Detection, Anomaly Detection, Time Series, Time Patterns, Data Size, Cyber-attacks, Cyber-hysical Systems, Critical Infrastructures, Industrial Control Systems

ACM Reference Format:
Muhammad Azmi Umer, Aditya Mathur, and Muhammad Taha Jilani. 2022. Effect of Time Patterns in Mining Process Invariants for Industrial Control Systems: An Experimental Study. In *Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (SEA4DQ '22), November 17, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3549037.3561274

1 INTRODUCTION

Industrial Control System (ICS) is a type of Cyber-physical Systems (CPS). It consists of cyber and physical components. Cyber components includes computing, and communication links while physical components are consist of sensors, actuators, and physical processes. These systems are integral part of many critical infrastructures. ICS is known to be vulnerable against cyber-attacks as evidenced by, for example, Stuxnet [1], Maroochy Water Services [2], and others. Researchers have made significant contributions to protect ICS against similar cyber-attacks. For this purpose, both traditional Intrusion Detection Systems (IDS), and those based on Machine Learning (ML), have been developed [3]. The study reported here focuses on ML-based IDS.

ML-based IDS often rely on the operational data obtained from a critical infrastructure. Several public datasets are available to create and evaluate including SWaT [4], EPIC [5], and WADI [6]. These datasets have been collected from operational testbeds that mimic the processes in city-scale plants. Since the datasets are collected by running the plant continuously over a period of time and hence contain multiple time series– one for each process variable. The study reported here, focuses on the question of how such time series can be used to determine the optimal size of the data to use in the creation of an IDS. Therefore, this work investigates the effectiveness of using variable chunks of multiple time series in mining the process invariants that capture the normal behavior of a plant when in a given state.

Several studies discuss the use of invariants [7–10] for detecting intrusions and process anomalies. However, there remains a significant gap in determining the effect of time patterns in mining the process invariants. The studies mentioned earlier do not assess the effect of time patterns on process invariants. In the study reported here, we assess the effects and usefulness of time patterns in mining the process invariants. It was observed that ignoring this property may lead to reduced effectiveness of the anomaly detector thus enabling attackers, with an exhaustive knowledge of plant dynamics, exploit the states embedded in overlooked invariants. Moreover, using the time patterns we can also determine the optimal data collection size for mining the process invariants. The creation of an effective ML-based IDS requires the right amount of data to avoid overfitting or underfitting issues. In the study reported here, we found that by using a suitable chunk-size based on the duration of a chunk, one could obtain the optimal data size for mining process invariants. The key objectives of the study are captured in the following research questions.

RQ1: How do time patterns affect mining the process invariants?
RQ2: How to determine optimal data collection size for generating process invariants?

SEA4DQ '22, November 17, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9459-8/22/11…$15.00
https://doi.org/10.1145/3549037.3561274

Contributions: (a) Demonstration of time pattern effects on invariants mining. (b) A proposal to determine optimal data collection size for generating process invariants.

Organization: The remainder of this paper is organized as follows. Section 2 describes the SWaT and the dataset. The methodology of invariant mining is described in Section 3. The metrics for optimal size of data are defined and discussed in Section 4. The experiments conducted using different datasets are discussed in Section 5. The research questions are discussed in Section 6. Section 7 discusses the related work and Section 8 contains the conclusion of the experimental study.

Figure 1: SWaT Testbed

2 SWAT TESTBED AND DATASET

The Secure Water Treatment (SWaT) shown in figure 1 is a testbed available at iTrust, Singapore University of Technology and Design (SUTD) [11]. It is an industrial replica and a scaled-down version of a water treatment plant. It is composed of six well-defined processes. The first stage involves the treatment of raw water. Chemical dosing and ultra-filtration are done at stages 2 and 3. Dechlorination is done at stage 4. After that water is passed to stage 5 to perform the reverse osmosis. The last stage distributes the water and also performs the backwash. SWaT can produce treated water at the capacity of 5 gallons per minute. The plant is composed of several sensors and actuators. Sensors are used to measure the level of water in tanks, to measure the flow of water, etc. Actuators include motorized valves and electric pumps. The communication between PLCs and sensors/ actuators is done using Level 0 network while Level 1 network is used for communication among PLCs.

The SWaT dataset [4] was collected by running the plant continuously in the normal state for seven consecutive days. There is a large number of studies that have reportedly used the SWaT dataset [12–15]. The SWaT dataset contains 51 attributes. In the current study, only 14 attributes which are described in Table 1 have been used. The Association Rule Mining (ARM) [16] approach used in the current study works only on binary-valued attributes. The SWaT dataset consists of binary, ternary, and real-valued attributes. Therefore we transformed all the dataset into binary-valued attributes as described in listing 1. Doing so resulted in a drastic reduction of attributes. After transformation into binary-valued attributes, most of the attributes were giving a single constant value throughout the dataset. This constant value was not useful for rule mining therefore these attributes were dropped from the dataset. Similarly, some attributes were not changing their values throughout the dataset like P102 because it is an electric pump that is for backup of P101. These types of backup attributes were also dropped from the dataset.

```
# MV-101 is the inlet valve controlled by a PLC depending
      on LIT101 measurements.
# FIT-101 is use to measure the flow towards Tank T101.
if FIT-101<0.5:
    FIT-101=0 # It means no flow
else:
    FIT-101=1 # It means there is a flow
if MV-101 == Open:
    MV-101 = Open
elif MV-101 == Close:
    MV-101 = Close
elif MV-101 == Transition:
    if FIT-101<0.5:
        MV-101 = Close
    else:
        MV-101 = Open

```

Listing 1: Feature Transformation into Binary-Valued Attributes

Algorithm 1: Invariant generation based on time intervals

1: $Dataset \leftarrow Dataset()$
2: $BinaryValued\_Attributes \leftarrow FeatureTransformation(Dataset)$
3: $Selected\_Features \leftarrow FeatureSelection(BinaryValued\_Attributes)$
4: $Time\_Interval \leftarrow Number\_of\_hours$
5: $Data\_Chunks \leftarrow DataChunks(Selected\_Features, Time\_Interval)$
6: $Support \leftarrow S$
7: $Confidence \leftarrow C$
8: **while** Data_Chunks is not empty **do**
9: $datachunk \leftarrow Data\_Chunks.pop()$
10: $Frequent\_Itemsets \leftarrow FrequentItemsets(datachunk, support)$
11: $Invariants \leftarrow AssociationRules(Frequent\_Itemsets, confidence)$
12: **end while**

3 INVARIANT MINING

Invariants represent the normal behavior of a physical plant. It is a condition that holds when the plant is in a given state. There could be various types of invariant as described in equation 1 and 2. Equation 1 represents an invariant which generates an alert when water level in Tank 101 goes below the LL (low-low) marker. Equation 2 represents an invariant which ensures that if water in Tank 101 is equal to or below the L (low) marker then Motorized Valve 101 should be open, else it generates the alert. In the current

work we have mined invariants of type (2) using Association Rule Mining (ARM) [16].

$$LIT101(k) > LL \tag{1}$$

$$LIT101 \leq L \implies MV101 = OPEN \tag{2}$$

Association Rule Mining. ARM [16] is an unsupervised rule-based machine learning approach. It is used to uncover the relationships between seemingly unrelated data in databases. This relationship is expressed as a rule such as P101=ON $\implies$ MV201=ON. The item to the left of $\implies$ is referred to as *antecedent* and the right one as the *consequent*. There are two major processes involved in ARM. The first one is frequent item generation and the second one is rule generation. Both of these processes are described below:

Frequent item sets. An itemset is a collection of values of one or more attributes e.g. state of pump, state of the motorized valve, etc. Every attribute with each possible value is considered as an itemset. Given the snapshot of the transformed dataset of SWaT at Figure 2, e.g. MV101=1 is an itemset, similarly, MV101=0 is another itemset. The itemsets which satisfy the minimum support threshold are considered as frequent itemsets. The support for an itemset A can be calculated using the following equation:

$$\text{Support}(A) = \frac{|e \in D; A \in e|}{|D|} \tag{3}$$

Here e is a transaction which exist in dataset D, and $|D|$ denotes the total number of transactions (rows) in the dataset.

Association rules. Once the frequent itemsets are obtained then they could be partitioned in more than one way to generate association rules, e.g. X $\implies$ Y, where X is referred to as antecedent, and Y is referred to as consequent. The rule which satisfies the minimum confidence criteria is considered as an association rule. The confidence of a rule can be calculated the following equation:

$$\text{Confidence}(X \implies Y) = \frac{Support(X \cup Y)}{Support(X)} \tag{4}$$

There could be multiple parts at Antecedent like X1, X2, X3 $\implies$ Y, where , is representing the Boolean *AND*. The complete algorithm is described in Algorithm 1

4 EVALUATION METRICS FOR OPTIMAL SIZE OF DATA

The following metrics are used to evaluate the optimal data collection size for mining process invariants.

4.1 Average sum of difference among similar sized Chunks

This metric sums up the differences in the number of invariants among each similar size chunks and then takes its average to obtain the Average Sum of Difference. It can be mathematically expressed as:

$$ASDSSC = \frac{\sum_{i=1}^{TNC-1} |NIC(i) - NIC(i+1)|}{TNC - 1} \tag{5}$$

where,
ASDSSC = Average Sum of Difference Among Similar Size Chunks,
TNC = Total Number of Similar Size Chunks, and
NIC = Number of Invariants in Chunk.

Table 1: Attributes selected in the current study

Attribute	Description
Flow meters	
FIT101	Measures inflow into tank T101
FIT201	Measures flow rate from stage1 to 2
FIT301	Measures the flow of water in the UF stage
Motorized valves	
MV101	Controls water flow into tank T101
MV201	Controls flow into tank T301
MV301	Controls the UF-backwash process
MV302	Controls water flow to the de-chlorination unit
MV303	Controls UF backwash
MV304	Controls UF backwash drain
Pumps	
P101	Pumps water from raw water tank to stage 2
P203	Dosing pump for HCl*
P205	Dosing pump for NaOCl*
P302	Pumps water from tanks T301 to T401
P602	Pumps water from backwash tank T602 to UF

*HCL and NaOCl are chemicals added to water at stage 2.

4.2 Standard deviation of similar sized Chunks

This metric determines the standard deviation across the number of invariants in similar size chunks. It can be mathematically expressed as:

$$SDSSC = \sqrt{\frac{\sum_{i=1}^{TNC}(NIC(i) - \frac{\sum_{j=1}^{TNC} NIC(j)}{TNC})^2}{TNC}} \tag{6}$$

where,
SDSSC = Standard Deviation of Similar Size Chunks,
TNC = Total Number of Similar Size Chunks, and
NIC = Number of Invariants in Chunk.

4.3 Average number of invariants in similar sized Chunks

This metric determines the average number of invariants in similar size chunks. It can be mathematically expressed as:

$$ANISSC = \frac{\sum_{i=1}^{TNC} NIC(i)}{TNC} \tag{7}$$

where,
ANISSC = Average Number of Invariants in Similar Size Chunks,
TNC = Total Number of Similar Size Chunks, and
NIC = Number of Invariants in Chunk.

4.4 Average number of common invariants in similar sized Chunks

This metric first determines the sum of the number of common invariants in adjacent similar size chunks and then calculates its

FIT101	MV101	P101	FIT201	MV201	P203	P205	FIT301	MV301	MV302	MV303	MV304	P301	P602
0	0	1	1	1	1	1	1	0	1	0	0	1	0

Figure 2: A snapshot of transformed dataset of SWaT with selected features

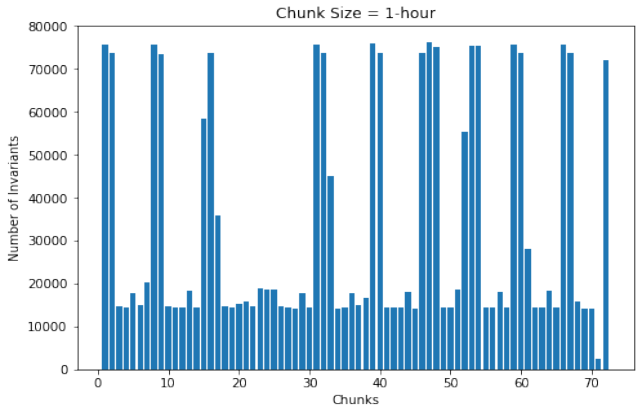


Figure 3: Change in the number of invariants in consecutive 1-hour chunks

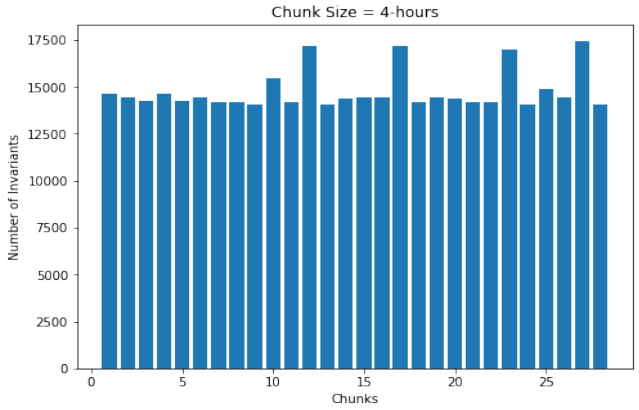


Figure 4: Change in the number of invariants in consecutive 4-hours chunks

average. It can be mathematically expressed as:

$$ANCISSC = \frac{\sum_{i=1}^{TNC-1} |I(i) \bigcap I(i+1)|}{TNC - 1} \quad (8)$$

where,
ANCISSC = Average Number of Common Invariants in Similar Size Chunks,
TNC = Total Number of Similar Size Chunks, and
I = Set of Invariants.

Discussion on Metrics. The metrics defined in Sections 4.1, 4.2, and 4.3, are related to the quantitative aspects of the invariants. They mainly focus on the stability in the number of invariants. At the beginning these metrics, evaluated over different chunk sizes, lead to stability with respect to the number of generated invariants. The metric in Section 4.4 is related to the qualitative aspect of the invariants. The earlier three metrics reveal the quantitative behaviour of invariants. However, there exists a possibility that while the number of invariants might be the same, indicating stability, though the invariants generated might be significantly different than those derived earlier. Metric ANCISSC in Section 4.4 is used to evaluate this qualitative aspect of the invariants. ANCISSC offers additional insights into the number of *common* invariants over different chunk sizes.

5 EXPERIMENTS

Experiments were conducted with two datasets collected from SWaT [11] during 2015 and 2020 containing, respectively, 410400 and 18000 rows. All experiments used the same number of attributes, support, and confidence threshold.

5.1 Effects of time patterns on process Invariants

5.1.1 Experiments using the SWaT 2015 Dataset. In the begining a subset containing 72 hours of 2015 dataset was selected (later we have used the complete 2015 dataset as shown in Figure 7). Next, this dataset was split into 72 independent chunks each comprising of one hour of data. Invariant mining was carried out independently on all chunks. The variation in the number of invariants obtained across the chunks is plotted in Figure 3. A pattern consisting of long and short bars can be discerned from the figure in the number of invariants generated. The question of interest was to discover the invariants that resulted in the long bars.

Upon deeper examination of the invariants in long and short bars, it was discovered that long bars contain invariants where the valve MV101 is open and there is some flow indicated by the flow rate indicator FIT101. Invariants related to the aforementioned actuators and sensors were not found in the short bars. To understand such discrepancy, the 72-hour chunk was next split into multiple 4-hour chunks and invariants mined again.

Figure 4 is a plot of the number of generated invariants using the 4-hour chunks. As shown, the longer bars in Figure 3 are now absent. After analyzing the invariants generated using 4-hours chunks, it was discovered that the invariants related to MV101-Open, and a non-zero reading from FIT101, were present in every bar in Figure 4. Thus, it was realized that further increase in the chunk size may affect the distribution of MV101-Open and non-zero FIT101 in the bars and may create additional stability in the number of invariants generated. For this purpose, chunks of size 7, 10, and 13-hours were created from the dataset as shown in Figure 6. It can be observed from this figure that the long bars gradually vanish as the chunk size is increased from 4 to 13, completely vanishing in Figure 6(d). It

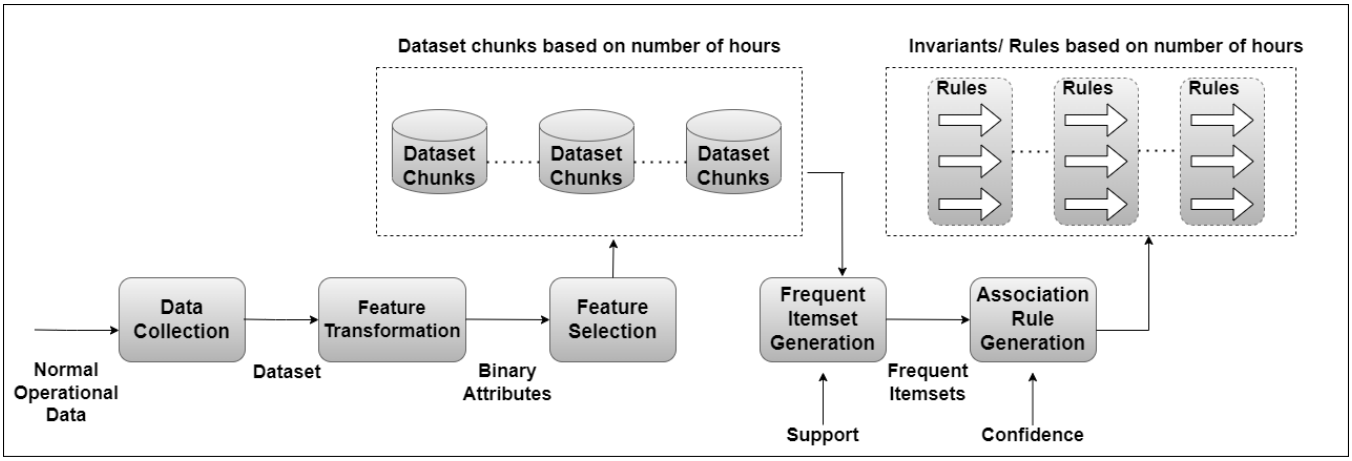


Figure 5: Invariant generation based on number of hours

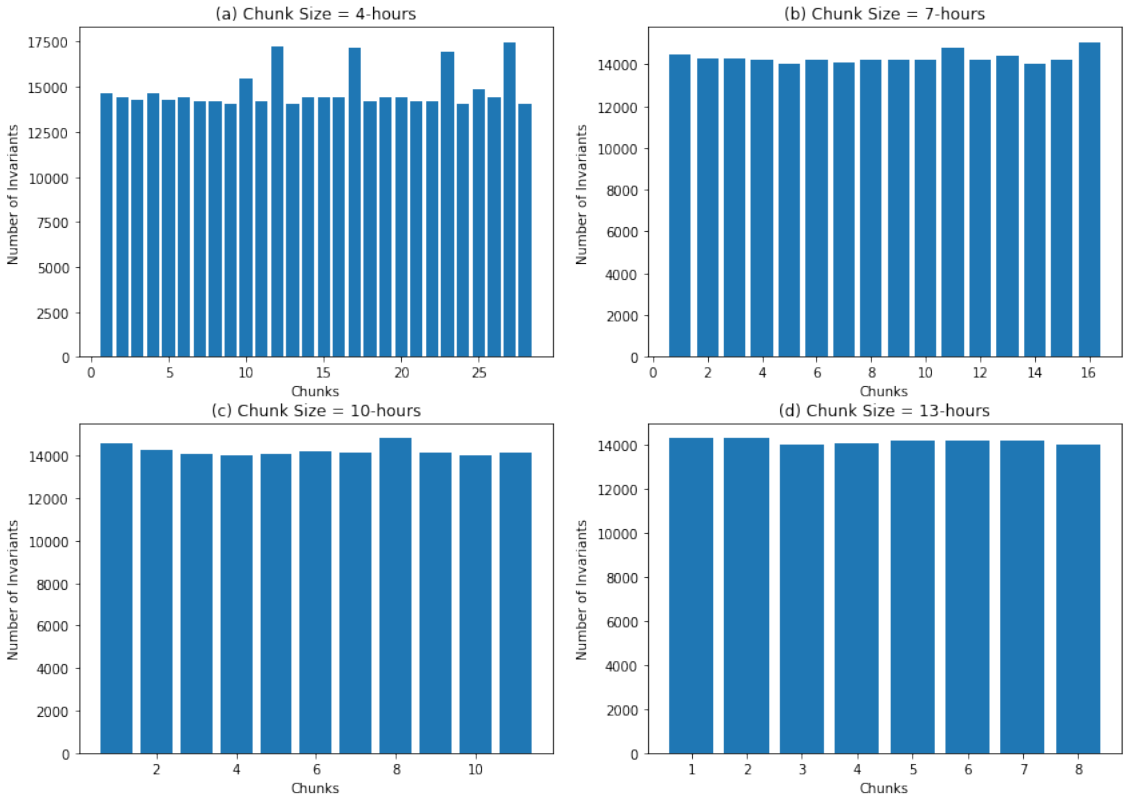


Figure 6: Change in the number of invariants with chunk duration in SWaT 2015 dataset.

is apparently clear from Figure 6 that our hypothesis, i.e., stability in the number of invariants generated with an increase in chunk size of the selected time series, holds.

5.1.2 Experiments using the SWaT 2020 Dataset. As the 2020 dataset of SWaT consists of 18000 rows, i.e., 5-hours of data, this dataset was split into 10 independent chunks each comprising of 30-minutes of data. Invariant mining was done independently on these chunks. We obtained a different number of invariants on these 10 chunks as shown in Figure 8(a). It can be observed from the figure that there is no pattern in the number of generated invariants as observed earlier in Figure 3. Next, we gradually increased the chunk size and created chunks based on 1, 1.5, 2, and 2.5-hours. This resulted in a well-defined pattern in the number of generated invariants as

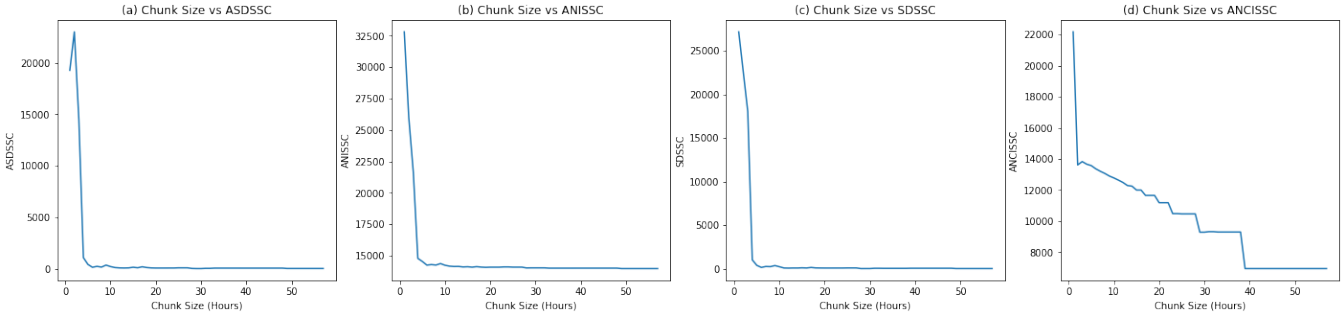


Figure 7: Change in metrics from Section 4 with chunk duration in SWaT 2015 dataset.

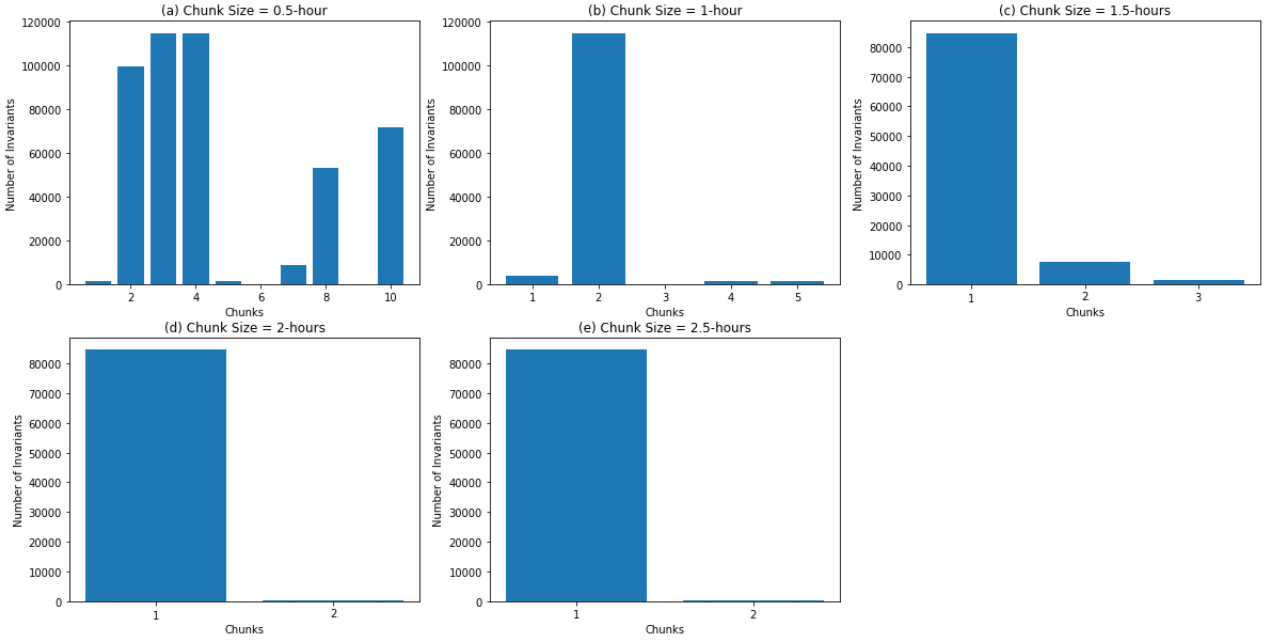


Figure 8: Change in the number of invariants with chunk duration in SWaT 2020 dataset.

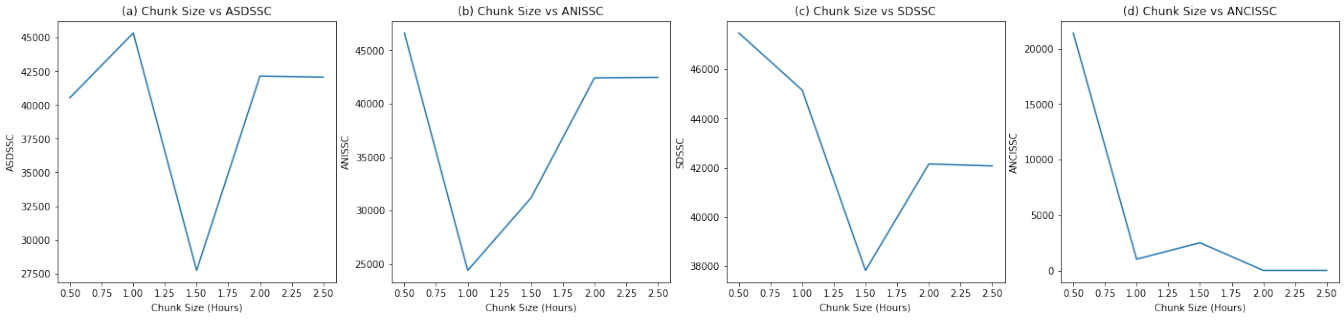


Figure 9: Change in metrics from Section 4 with chunk duration in SWaT 2020 dataset.

shown in Figures 8(c), (d), and (e). This observation reaffirms our hypothesis, i.e., number of invariants generated gradually becomes stable with an increase in chunk size of the selected time series.

5.2 Optimal data size to generate Invariants

5.2.1 Experiments using SWaT 2015 Dataset. The proposed approach for optimal size of data to be collected was evaluated using

metrics defined in Section 4. For this purpose, all possible chunk sizes were created based on the number of hours in the entire dataset as shown in Figure 5. The minimum and maximum data chunk sizes were, respectively 1 and 57 hours. The results from all the experiments based on the aforementioned metrics are presented in Figure 7. These metrics are designed to determine the stability in the number of generated invariants over different time intervals.

5.2.2 Experiments using SWaT 2020 Dataset. The sizes of data chunks created from the SWaT 2020 dataset were 0.5, 1, 1.5, 2, and 2.5 hours. The results from all the experiments based on the aforementioned metrics are presented in Figure 9. As these metrics are designed to determine the stability of the generated invariants over different time intervals, it is evident from Figure 9 that chunk sizes greater than 2-hours lead to a stable set of invariants.

6 DISCUSSION

We next revisit the research questions stated earlier.

RQ1: Time patterns have a significant effect on the mining of process invariants. In the experiments described above, a significant difference was observed particularly in the number of invariants based on specific time patterns. This difference is due to specific sub-processes in SWaT that are activated at different intervals during plant operation. Some intervals include a large number of invariants while others include significantly less. The intervals with a large number of invariants exist for a specific duration. Therefore, the invariants mined in such intervals exist only in those intervals. No such behavior is observed in the remaining operation of the plant. The question arises whether one should consider such invariants to monitor process anomaly? The answer is in the affirmative because an attacker with an exhaustive knowledge of plant dynamics can exploit relations embedded in such invariants. Discarding such invariants as monitors for process anomaly would enable an attacker to specifically perform targeted attacks in those intervals and remain undetected.

RQ2: This question focuses on the optimal data size for mining process invariants. This optimal data size should capture the overall behavior of the ICS. Currently, we are unaware of any such study that focuses on this issue. For example, there exist classifiers that work quite well even on small-sized datasets, e.g., Naive Bayes. Similarly, there exist classifiers that require a large amount of training data to perform precise classification, e.g., Decision Trees. Though here we have particularly focused on optimal data size for mining process invariants, we believe this optimal data size would also be useful for IDS created using machine learning techniques. Since the invariants capture the normal behavior of an ICS, therefore, if we are able to determine the optimal data size for invariants then it would also be useful for ML-based IDS. From the plots in Figure 7 and 9, it is evident there is stability in the graphs after the chunk size reached 13-hours and 2-hours in SWaT 2015 and 2020 datasets, respectively. Therefore, we claim that for the SWaT 2015 and 2020 datasets, the optimal data size is 13 and 2-hours, respectively. This data size should capture the overall behavior of SWaT.

7 RELATED WORK

ARM was used in [9, 17] to generate invariants for ICS, with operational data from SWaT [4], to mine the invariants. These studies ignored the time series property of SWaT data. The study proposed here exploited the time series property of SWaT data to study the effects of time patterns in mining the process invariants. A study reported in [18] used the ARM to generate the attack patterns for an ICS. For this purpose, they used both the attacked and normal operational data of SWaT. However, they also ignored the time series property of SWaT data.

Process invariants were generated for a selected sub-process of SWaT using its design information in [19]. They evaluated their approach by performing different attacks on the testbed. The proposed approach was found useful in detecting the attacks, however, it does not exploit the time-series property of SWaT. A study reported in [20] used a set of machine learning approaches to generate invariants for two real-world ICS. They considered the time series property for mining the invariants. However, they did not discuss the usefulness of time patterns in mining the process invariants as in the current study. Supervised Machine learning technique was used in [21] to generate process invariants for SWaT. They also considered the time series property of SWaT data. However, they did not report the effect of time patterns in mining the process invariants.

8 CONCLUSION

The evolution of each state variable controlled by an ICS in a critical infrastructure can be represented as a time series. The study reported here highlights the value of such time series in mining process invariants, i.e., those that capture the plant behavior under normal conditions. It was observed that ignoring this property may result in missed invariants leading to an ineffective anomaly detector and thus enabling an attacker, with an exhaustive knowledge of plant physics, to exploit the condition embedded in the lost invariants. The attacker could potentially target such time intervals that contain the lost invariants. Secondly, the creation of effective ML-based IDS requires the right amount of data to avoid overfitting or underfitting. In the current study we found that by using a suitable chunk size based on the duration of a chunk, one could obtain the optimal data size for mining process invariants.

ACKNOWLEDGEMENTS

This research is supported by the National Research Foundation, Singapore, under its National Satellite of Excellence Programme "Design Science and Technology for Secure Critical Infrastructure" (Award Number: NRF2018NCR-NSOE005-0001). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

[1] N. Falliere, L.O. Murchu, and E. Chien. W32 stuxnet dossier. symantec, version 1.4. https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier.pdf, 2 2011.
[2] M. Abrams and J. Weiss. Malicious control system cyber security attack case study–Maroochy Water Services, Australia. Technical report, The Mitre Corporation, McLean, VA, August 2008. http://csrc.nist.gov/groups/SMA/fisma/ics/documents/Maroochy-Water-Services-Case-Study_briefing.pdf.

[3] Muhammad Azmi Umer, Khurum Nazir Junejo, Muhammad Taha Jilani, and Aditya P Mathur. Machine learning for intrusion detection in industrial control systems: Applications, challenges, and recommendations. *International Journal of Critical Infrastructure Protection*, 38:100516, 2022.
[4] iTrust. Dataset and models. https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/, 2021.
[5] Chuadhry Mujeeb Ahmed and Nandha Kumar Kandasamy. A comprehensive dataset from a smart grid testbed for machine learning based cps security research. In *International Workshop on Cyber-Physical Security for Critical Infrastructures Protection*, pages 123–135. Springer, 2020.
[6] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. Wadi: A water distribution testbed for research in the design of secure cyber physical systems. In *CysWater*, pages 25–28, NY, USA, 2017. ACM.
[7] Koyena Pal, Sridhar Adepu, and Jonathan Goh. Effectiveness of association rules mining for invariants generation in cyber-physical systems. In *Proceedings of the 18th IEEE Symposium on High Assurance Systems Engineering*, pages 124–127, Washington, D.C., USA, January 2017. IEEE Computer Society.
[8] Chuadhry Mujeeb Ahmed, Muhammad Azmi Umer, Beebi Siti Salimah Binte Liyakkathali, Muhammad Taha Jilani, and Jianying Zhou. Machine learning for cps security: Applications, challenges and recommendations. In *Machine Intelligence for Cybersecurity Applications*, pages 397–421. Springer, 2021.
[9] Muhammad Azmi Umer, Aditya Mathur, Khurum Nazir Junejo, and Sridhar Adepu. Generating invariants using design and data-centric approaches for distributed attack detection. *IJCIP*, 28:100341, 2020.
[10] Danish Hudani, Muhammad Haseeb, Muhammad Taufiq, Muhammad Azmi Umer, and Nandha Kumar Kandasamy. A data-centric approach to generate invariants for a smart grid using machine learning. In *Proceedings of the 2022 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems*, pages 31–36, 2022.
[11] A. P. Mathur and N. O. Tippenhauer. SWaT: A water treatment testbed for research and training on ICS security. In *International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*, pages 31–36, USA, April 2016. IEEE.
[12] Khurum Nazir Junejo. Predictive safety assessment for storage tanks of water cyber physical systems using machine learning. *Sādhanā*, 45(1):1–16, 2020.
[13] Khurum Nazir Junejo and Jonathan Goh. Behaviour-based attack detection and classification in cyber physical systems using machine learning. In *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*, CPSS '16, page 34–43, New York, NY, USA, 2016. Association for Computing Machinery.
[14] Dušan M Nedeljković, Živana B Jakovljević, Zoran Đ Miljković, and Miroslav Pajić. Detection of cyber-attacks in systems with distributed control based on support vector regression. *Telfor Journal*, 12(2):104–109, 2020.
[15] Astha Garg, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, and Chuan-Sheng Foo. An evaluation of anomaly detection and diagnosis in multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
[16] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *ICMD*, volume 22, pages 207–216, New York, NY, USA, June 1993. ACM.
[17] Muhammad Azmi Umer, Aditya Mathur, Khurum Nazir Junejo, and Sridhar Adepu. Integrating design and data centric approaches to generate invariants for distributed attack detection. In *Proceedings of the 2017 workshop on cyber-physical systems security and privacy*, pages 131–136, 2017.
[18] Muhammad Azmi Umer, Chuadhry Mujeeb Ahmed, Muhammad Taha Jilani, and Aditya P Mathur. Attack rules: an adversarial approach to generate attacks for industrial control systems using machine learning. In *Proceedings of the 2th Workshop on CPS&IoT Security and Privacy*, pages 35–40, 2021.
[19] S. Adepu and Aditya Mathur. Using process invariants to detect cyber attacks on a water treatment system. In *Proceedings of the 31st International Conference on ICT Systems Security and Privacy Protection - IFIP SEC 2016 (IFIP AICT series)*, pages 91–104, New York, USA, 2016. Springer.
[20] Cheng Feng, Venkata Reddy Palleti, Aditya Mathur, and Deeph Chana. A systematic framework to generate invariants for anomaly detection in industrial control systems. In *NDSS*, 2019.
[21] Yuqi Chen, Christopher M. Poskitt, and Jun Sun. Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system. *IEEE Security and Privacy 2018*, abs/1801.00903, 2018.

Preliminary Findings on the Occurrence and Causes of Data Smells in a Real-World Business Travel Data Processing Pipeline

Valentina Golendukhina
University of Innsbruck
Austria
valentina.golendukhina@uibk.ac.at

Harald Foidl
University of Innsbruck
Austria
harald.foidl@uibk.ac.at

Michael Felderer
University of Innsbruck
Austria
michael.felderer@uibk.ac.at

Rudolf Ramler
Software Competence Center Hagenberg GmbH
Austria
rudolf.ramler@scch.at

ABSTRACT

Detection of poor quality data is crucial for enhancing data-driven systems' quality. Although there is a lot of research on data validation, the topic of potential data quality issues is still underexplored. Such latent issues or *data smells* can often stay undetected and lead to the poor future performance of data-intensive systems. Detecting data smells is not trivial and requires knowledge about their causes. In this paper, we present the preliminary findings on the causes and severity of data smells based on a study of a real-world business travel data set and the data processing pipeline behind it. The results show that data smells exist in this data set and cause severe problems. Although many data smells already occur in raw data, some smells are created during the transformation and enrichment stages of the data processing pipeline. These findings indicate the importance of the data pipeline itself for future research on data smells. Thus, this article proposes potential future work in this area.

CCS CONCEPTS

• **Information systems** → **Information integration**.

KEYWORDS

Data smells, data pipeline, data issues

ACM Reference Format:
Valentina Golendukhina, Harald Foidl, Michael Felderer, and Rudolf Ramler. 2022. Preliminary Findings on the Occurrence and Causes of Data Smells in a Real-World Business Travel Data Processing Pipeline. In *Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (SEA4DQ '22), November 17, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3549037.3561275

1 INTRODUCTION

With the wide spread of machine learning and data-intensive systems, data quality becomes a more and more critical issue. Various systems rely on large amounts of data for different purposes from performing business analyses to facilitating the decision-making process. Overall, technologies based on data provide valuable solutions to existing problems given that the data used for such purposes is of high quality.

In practice, using data of poor quality can lead to unpredictable consequences and financial reputation, or even human loss. The lack of common standards for data collection and preparation methods makes it difficult to apply a unified solution to every problem. However, the adoption of data pipelines automates the data flow process to improve the final data quality. A data processing pipeline is a sequence of operations with data including data ingestion, integration, cleaning, transformation, enrichment, and loading [5], as well as functionalities for data flow monitoring and management. Although data pipelines enhance productivity and contribute to the quality of data [10], poorly developed and buggy data pipelines may not only fail to recognize data quality issues but also produce data of poor quality [12]. Such data issues can stay unnoticed through all processing operations and lead to incorrect results in the future. Identification and early prevention of the latent and context-independent data quality issues, i.e., data smells [4], is crucial to provide data of high quality and achieve better results.

In this paper, we investigate to what extent data pipeline elements may lead to the creation of data smells and what are the possible consequences of it based on business travel data. The business travel industry generates a significant amount of data [11] that includes transportation and accommodation stages. Although the big amounts of data are not new to the industry, the wide spread of information and communications technologies and different standards inherent in them lead to high heterogeneity in raw data [2]. Thus, data processing is complex and prone to errors.

As business travel data are characterized by data describing geographical coordinates such as longitude and latitude, we argue that the analysis and findings in this paper are highly relevant to the domain of Internet of Things (IoT), in particular for autonomous vehicles, drone navigation, and other domains dependent on location, time, and distance calculations. Moreover, travel data have a large potential for sustainable solutions testing and application regarding carbon footprint calculations, management, and reduction, which is highly relevant for any industry nowadays.

The aim of this exploratory study is threefold: first, we want to understand to which extent data are affected by data smells; second,

SEA4DQ '22, November 17, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9459-8/22/11.
https://doi.org/10.1145/3549037.3561275

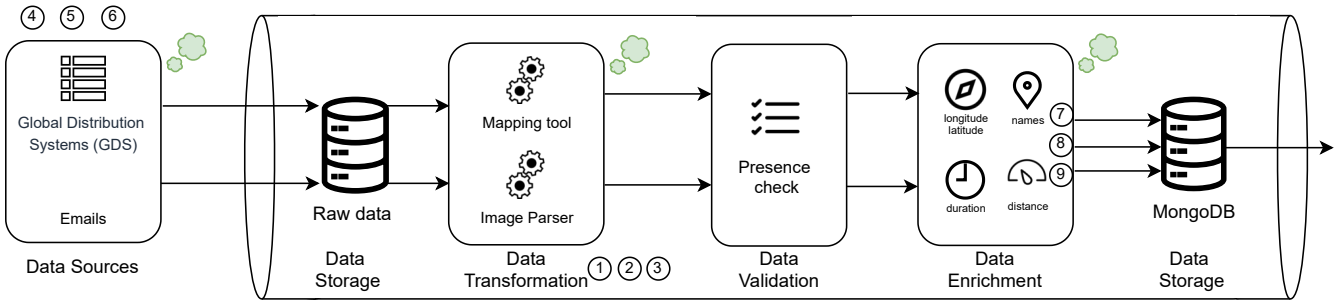


Figure 1: Overview of the main components of the data pipeline

we have a closer look at the roots of data smells and their place in the data pipeline; third, we propose the ways to detect and avoid data smells in the data engineering process. Furthermore, with the results of this study, we pave the way for a deep quantitative study to investigate the causes and effects of data-related issues.

The rest of the paper is structured as follows. Section 2 presents the concept of data smells and the reasons for them. In Section 3, data smells are shown in connection with the data processing pipeline based on the real-life data set. Finally, Section 4 describes the future work to identify and prevent data smells and concludes the paper.

2 CONCEPT OF DATA SMELLS

Various studies have been dedicated to data quality issues and the consequences of poor quality data for data-driven systems [7–9]. Furthermore, to overcome these consequences, tools that can detect potential issues have been developed [1, 6]. The problems of missing data, data outliers, or duplicated values affect the final quality, but they can be detected before the damage occurs. Unlike typical data issues, data smells usually do not correspond to "obviously" wrong or incorrect data. Similar to the code smells, they represent latent issues, that might arise in the future.

Foidl et al. [4] distinguish four main characteristics of data smells. Such data have *moderate degree of suspicion*, i.e., they are not considered poor upon initial inspection. For instance, the word "Paris" represents a city and does not look suspicious. However, problems may arise as soon as the exact location is needed since there are at least two cities named Paris, one in France and one in the USA. Another important characteristic of data smells is their *context independence*. They represent universal issues that can be present in any domain and affect any data-driven system. Furthermore, smelly data can stay unnoticed for a long time and *lead to problems in the future*, e.g., less precise classification algorithms or wrong descriptive statistics. Finally, data smells are frequently *caused by poor practices* in data management and engineering. Thus, the introduction of quality assurance methods improves the quality of the processed data.

3 CASE STUDY

To explore the impact of data smells in a real-life scenario, we investigated the business travel data of a large enterprise with over 100,000 employees. The data set combines information about more than 130,000 trips.

To manage the large amounts of heterogeneous data and to use it for further analysis, the company developed and implemented a data pipeline. To analyze the data smells and their impact, we examined the data set and the pipeline to locate the causes of the smells. Figure 1 presents an overview of the data flow within the company's data pipeline. Green clouds indicate the stages where new data are produced and, therefore, data smells might originate. The numbers indicate the data smells we identified, as discussed in the further sections.

There are six main stages of data processing: data ingestion from data sources to the data storage, storage of the raw data, transformation to the formats applicable for further processing, data enrichment, and loading of the data into the final database. The prepared data are utilized for reporting, analysis, and predictions.

The data comes from two main types of data sources: global distribution systems (GDSs) and emails. A GDS is a system that provides information from various travel industry service providers, including airlines, railways, hotels, and car rental agencies. The structure and context of data provided by different GDSs vary dramatically. Thus, it does not allow direct analysis of the provided data and requires several preprocessing activities. The second source of information is unstructured data retrieved from emails, e.g., receipt documents or tickets. To further use the data from these sources, it has to be extracted and converted into a structured form.

Data transformation includes two main activities: mapping of the data to a fixed schema and data extraction using various image parsers. Then, a validation tool checks if all necessary information was extracted and whether it is meaningful, for instance, start and end dates for flights and trains; otherwise, the data record does not proceed to the next phase. During the enrichment, longitudes and latitudes are assigned. Based on this information, distance, duration, and time zones are calculated. The last enrichment element is a normalized name. Finally, data are stored in the database.

3.1 Identified Data Smells

To identify data smells in the given data set, we used the data smells taxonomy proposed by Foidl et al. [4]. The taxonomy includes 36 data smells separated into four main categories: encoding, consistency, syntactic, and believability smells. While manually analyzing our dataset, we found data smells related to all categories. They are listed in Table 1. Based on the amount of data affected by the smell

Table 1: Data smells identified in the real-life data set

№	Issue	Data smell type	Data smell subtype	Severity
1	Different types of date/time data: datetime and timestamp	Encoding smell	*Date/Time as string*	medium
2	Numerical values represented as strings	Encoding smell	*Number as string*	medium
3	Null and NaN values in one column	Consistency smell	*Missing value inconsistency*	medium
4	"9999.99.99" as a missing value for dates	Consistency smell	*Missing value inconsistency*	low
5	Abbreviations used for locations	Syntactic smell	*Ambiguous value*	high
6	Ambiguous location names	Syntactic smell	*Homonyms*	low
7	Incorrect longitudes and latitudes	Believability smell	*Suspect value*	high
8	Incorrect distances	Believability smell	*Suspect value*	high
9	Incorrect duration values	Believability smell	*Suspect date/time interval*	high

and the degree of the complexity of the problem, we introduced the *severity* column that varies from low, to medium and high.

3.1.1 Encoding Smells. Encoding smells describe the problems connected with inappropriate data types. Such issues complicate the data analysis process and might lead to wrong results. We found two instances of this smell type in the data set.

Date/Time as string. The columns with information about the time of arrival and departure have different types. Some of the columns are converted to DateTime type with timestamp type values, whereas the others are assigned object type with DateTime values. By data ingestion, DateTime format is returned if different time zones are present in a column. Transformation of the column to the right format without considering different time zones can lead to the loss of important information. Furthermore, all operations with time and date can be affected and deliver poor results.

Number as string. Numerical values representing duration in minutes and IDs have string type. This could lead to complications if numerical operations are applied to the data. Also, there is a risk of information loss if transformed incorrectly (integer instead of string).

3.1.2 Consistency Smells. Consistency smells arise when different methods are used to resolve equivalent problems. As a result, the same data have nonidentical expressions, which makes further analysis inconsistent and prone to errors.

Missing value inconsistency. We found that missing values expressions vary within the data set and within one column. Two ways to represent missing values are *NaN* and *None*. Since they have different properties, it can affect the identification of missing values, and make operations with missing values more difficult. For instance, a comparison of None values returns True, whereas a comparison of NaN values returns False.

Another case of inconsistent missing values was found in date information, where "9999.99.99" represented a missing value. This representation was recognized as a date and did not account for missing values by the system. Since we only found 242 values represented in this manner, we classify this issue as low severity for the given data set.

3.1.3 Syntactic Smells. Syntactic smells represent inappropriate expressions of values that might lead to misinterpretation of information by humans or algorithms.

Ambiguous value. Several train stations are located in the airports. In some of these cases, train stations use the abbreviation of the airport as their name. However, such names for train stations were not recognized correctly and led to misinterpretation of locations and false classification of transportation methods (by train or by plane). In total, 2,922 items were affected by this problem. It worsened the quality of the data and affected the next steps of data processing, particularly the data enrichment process.

Homonyms. Nonunique departure and arrival names result in data smells that can stay unidentified for a long period. However, when used for longitude and latitude calculations, the quality of such operations is rather poor. In the data set, distances were calculated wrongly for trains with the departure or arrival in Boston. The name is considered ambiguous because there are two Bostons in USA and Australia. As in the case of ambiguous values, homonyms lead to errors in the data enrichment process.

3.1.4 Believability Smells. Believability smells can be interpreted by software as correct data and understood by humans, but they do not represent the right data. To identify these smells, a descriptive analysis is needed.

Incorrect longitudes and latitudes. Longitude and latitudes represent geographic coordinate systems and are necessary for distance calculations, mapping, and other purposes. The wrong values are difficult to identify because the values are represented as float numbers with several decimal digits depending on the precision of the algorithm. Wrong latitudes or longitudes lead to incorrect calculations of distance and duration.

Incorrect distances. As it is mentioned above, incorrect distances appear because of the wrong latitudes and longitudes. In case they are negative or extremely large, they can be detected more easily. However, if the distance is within an accepted range, the issue can stay unnoticed.

Incorrect duration values. The time difference between arrival and departure is calculated within the data pipeline. To estimate the duration correctly, departure and arrival times must be adjusted following local time zones. If the information about the time zone is lost or the time zone is miscalculated, the final result is affected.

3.2 Causes of Data Smells in the Data Pipeline

To understand the causes of the data smells, we investigated the data pipeline and identified the steps where data smells occur. The

results are shown in Figure 1. The enumeration of data smells done as it is stated in Table 1.

Data sources. Originally, some of the data smells come from the raw data. Some of the values can be written inconsistently because they are entered by different users. Another reason for inconsistent data is the variety of sources and differences in the methods of data management and handling. The data providers in the travel domain frequently improve and change their data schemes, therefore, the raw data have high variability depending on vendors.

Data transformation. The next group of data smells arises in the data transformation stage. Date/time as string issue arises because of the insufficient handling of time zones and time values, that represent empty values. Whereas some tools will recognize "9999-12-12" as a time variable, others recognize it as a string. Depending on the result, the type of column is assigned.

Number as a string is a smell caused by the differences in the processing of structured and unstructured data. The problem is inherent in the process of document parsing. All data parsed from images is assigned to string data type to avoid possible data loss.

The reason for missing values inconsistency is different methods and tools used during data transformation on different data. If tools use different programming languages, e.g., Java and JavaScript, the results of the operations will vary based on the logic of the programming language. In the case of this pipeline, the image parser is based on Java and the mapping tool is written in JavaScript.

Data enrichment. The majority of data smells are produced in the data enrichment phase. A part of them is due to smelly raw data or poor data transformation techniques. The creation of smells in the enrichment phase happens due to insufficient validation in the previous phases. However, data smells can also be produced from clean data, e.g., if time zones are not considered.

Moreover, this new data can be beneficial for data smells identification in the previous steps. Although spotting data smells in some cases might be challenging, analysis of the products of such data can facilitate the process.

3.3 Data Smells Detection

Data smell detection is not a trivial task that requires specific solutions. Although data smell detection is closely related to the data validation process, there are several reasons why it should be performed separately [4]. Firstly, detection tools can reveal many smells and produce a large number of alerts that are impossible to be processed effectively. Secondly, it requires a certain amount of processing capabilities and should not worsen the performance of AI-based systems.

So far, there are rule-based and ML-based tools for data smell detection. However, as we observed in the previous section, data smells appear in different data processing phases approaches might vary in their suitability and effectiveness based on the data pipeline stage. Understanding when the tools should be applied to achieve the best results would significantly improve the quality of the final data. Moreover, considering particular qualities of data pipelines would improve the quality in the long term.

4 CONCLUSIONS AND FUTURE WORK

With this exploratory study, we contribute to practical evidence of data smells and motivate several directions of future work. Based on the data smells taxonomy, we provide the evidence and highlight the widespread occurrence of data smells in real-life data sets and the threats they represent for data-driven systems. Furthermore, we outline the connection between data pipeline architecture and the creation of data smells. Additionally, we discussed the opportunities for improvement of data smell detection tools.

Inevitably, a data pipeline has a high impact on data quality. Therefore, the strategies developed for data validation should be adapted to the data pipeline architecture. Moreover, the adjustment of data validation practices to different stages of the data pipeline and the outcomes can be investigated further. Future research should study the impact of data smells in different domains and determine universal practices for data quality improvement.

Based on the first results presented in this paper, our future goals are to develop specific techniques that detect potential causes of data smells in pipelines and a method to determine the severity of data smells, i.e., how the severity of smells can be computed. Also, we want to identify pipeline patterns that are suspect in creating data smells and focus on data transformation and enrichment, the phases that caused the majority of smells. As for the data sources, we want to investigate the relationship between the quality of raw input data and the number of smells they produce [3]. All of these steps can further lead to the development of a quality model representing the quality characteristics of data pipelines.

ACKNOWLEDGMENTS

This work was supported by the Austrian ministries BMK & BMDW and the State of Upper Austria in the frame of the COMET competence center SCCH [865891], and by the Austrian Research Promotion Agency (FFG) in the frame of the projects Green Door to Door Business Travel [FO999892583] and ConTest [888127].

REFERENCES

[1] Daniel W Barowy, Dimitar Gochev, and Emery D Berger. 2014. Checkcell: Data debugging for spreadsheets. *ACM SIGPLAN Notices* 49, 10 (2014), 507–523. https://doi.org/10.1145/2714064.2660207
[2] Thomas H Davenport. 2013. turning towards a smarter travel experience. (2013).
[3] Harald Foidl and Michael Felderer. 2022. An Approach for Assessing Industrial Iot Data Sources to Determine Their Data Trustworthiness. (2022). https://doi.org/10.2139/ssrn.4069988
[4] Harald Foidl, Michael Felderer, and Rudolf Ramler. 2022. Data Smells: Categories, Causes and Consequences, and Detection of Suspicious Data in AI-based Systems. In *1st International Conference on AI Engineering – Software Engineering for AI (CAIN)*. IEEE/ACM, 229–239.
[5] Hannes Hapke and Catherine Nelson. 2020. *Building machine learning pipelines*. O'Reilly Media.
[6] Nick Hynes, D Sculley, and Michael Terry. 2017. The data linter: Lightweight, automated sanity checking for ml data sets. In *NIPS MLSys Workshop*, Vol. 1.
[7] Joao Marcelo Borovina Josko, Lisa Ehrlinger, and Wolfram Wöß. 2019. Towards a Knowledge Graph to Describe and Process Data Defects. *DBKDA 2019* (2019), 65.
[8] Lin Li, Taoxin Peng, and Jessie Kennedy. 2014. A rule based taxonomy of dirty data. *GSTF Journal on Computing (JoC)* 1, 2 (2014). https://doi.org/10.5176/978-981-08-6308-1_d-035
[9] Jianzheng Liu, Jie Li, Weifeng Li, and Jiansheng Wu. 2016. Rethinking big data: A review on the data quality and usage issues. *ISPRS journal of photogrammetry and remote sensing* 115 (2016), 134–142. https://doi.org/10.1016/j.isprsjprs.2015.11.006
[10] Aiswarya Raj Munappy, Jan Bosch, and Helena Homström Olsson. 2020. Data pipeline management in practice: Challenges and opportunities. In *International Conference on Product-Focused Software Process Improvement*. Springer, 168–184. https://doi.org/10.1007/978-3-030-64148-1_11
[11] Ben Vinod. 2016. Big data in the travel marketplace. *Journal of revenue and pricing management* 15, 5 (2016), 352–359. https://doi.org/10.1057/rpm.2016.30
[12] Haiyin Zhang, Luís Cruz, and Arie van Deursen. 2022. Code Smells for Machine Learning Applications. *arXiv preprint arXiv:2203.13746* (2022).

Data Quality Issues for Vibration Sensors: A Case Study in Ferrosilicon Production*

Maryna Waszak, Terje Moen, Sølve Eidnes — SINTEF AS, Norway
Alexander Stasik, Anders Hansen, Gregory Bouquet — SINTEF AS, Norway
Antoine Pultier, Xiang Ma — SINTEF AS, Norway
Idar Tørlen, Bjørn Henriksen, Arianeh Aamodt — Elkem ASA, Norway
Dumitru Roman — SINTEF AS, Norway

ABSTRACT

Digitisation in the mining and metal processing industries plays a key role in their modernisation. Production processes are more and more supported by a variety of sensors that produce large amounts of data that meant to provide insights into the performance of production infrastructures. In the metal processing industry vibration sensors are essential in the monitoring of the production infrastructure. In this position paper we present the installation of vibration sensors in a real industrial environment and discuss the data quality issues we encountered while using such sensors.

CCS CONCEPTS

• **Information systems → Information systems applications**.

KEYWORDS

vibration sensors, data quality, ferrosilicon production

ACM Reference Format:
Maryna Waszak, Terje Moen, Sølve Eidnes, Alexander Stasik, Anders Hansen, Gregory Bouquet, Antoine Pultier, Xiang Ma, Idar Tørlen, Bjørn Henriksen, Arianeh Aamodt, and Dumitru Roman. 2022. Data Quality Issues for Vibration Sensors: A Case Study in Ferrosilicon Production. In *Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (SEA4DQ '22), November 17, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3549037.3561273

1 INTRODUCTION

Within mining and metal processing industries, digital transformation is becoming a driving force changing the nature of companies and interaction with employees, communities, government, and environment at every step of the value chain [1]. The metal processing industry is already gathering a huge amount of data from sensors to collect real-time information about the performance of their infrastructure. Since many processes and machines can possibly generate data, smart sensors become a primary data source for producing insights via big data analytics. There remain however many areas where the industry lacks necessary and real-time information. Commercial sensor equipment may be available but could be too expensive or inadequate for direct implementation in the process. In addition, conditions related to the hostile nature of many processes, e.g., high temperature, dust, abrasion, corrosion, etc., may render data acquisition challenging. Research is thus needed to identify, evaluate, or develop sensor technologies to be used for real-time data gathering in harsh environments. Before smelting in metal processing, crushing, and sieving of raw materials are crucial process steps as raw materials have a large impact on the efficiency of the smelting process. Implementation of practical and reliable technologies monitoring such equipment in real-time will thus enable an improved optimisation of the smelting process. Furthermore, for the crushing and sieving process of the raw material, there is much to gain by optimising the process. Currently very little data is collected, except for the final product, which is too late to be used for process optimisation.

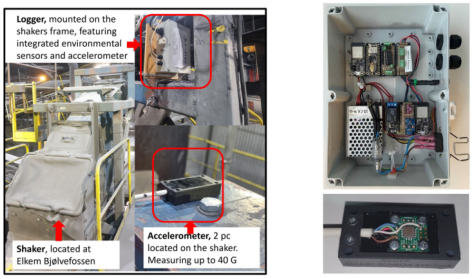


Figure 1: Left: Vibration sensing installation in ferrosilicon crushing facility at Elkem Bjølvefossen, Norway. Right: A vibration sensor encapsulated in a 3D printed watertight package with wiring together with an overview from inside the logger enclosure that contains the ESP43 WROOM boards and the power supply unit.

A case study was developed by Elkem – one of the world's leading providers of advanced material solutions – to explore vibration monitoring of mechanical sieving equipment for fault detection. The task focuses on developing suitable sensors to monitor the sieve screens in the material separators at Elkem Bjølvefossen, Norway plant with the goal to detect overfeeding and increase of the production throughput. A set of linear accelerometers were installed at selected positions on the separator and the vibration data is being collected since April 2022 (Fig. 1). Several data quality issues arise during data collection. In this position paper, we present the data acquisition pipeline and the issues that arise in the context of data

*This work received partial funding from the projects: BigDataMine (NFR 309691, MOST 2019YFE0105000), DataCloud (H2020 101016835), and SINTEF SEP - DataPipes.

SEA4DQ '22, November 17, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9459-8/22/11.
https://doi.org/10.1145/3549037.3561273

quality assessment. In our set-up we experience data loss of 8.6% due to the chosen acquisition strategy that we will explain here in more details and propose a possible mitigation strategy.

2 DATA ACQUISITION PIPELINE AND DATA QUALITY ISSUES

2.1 Hardware Set-Up

As depicted in Fig. 1 the data acquisition is performed with the following hardware that was installed on site in the crushing and sieving facility:

- Two three-axis ADXL356 accelerometers
- ESP32 WROOM-32E
- Lenovo Thinkbook PC with Windows 10
- tp-link Archer MR 600 router

The set-up is custom made in-house and installed in a separate network to ensure full control over the acquisition pipeline. Special attention was given to design a dust and watertight encapsulation of the equipment to prevent damage during the wet cleaning of the facility and to ensure long-lasting run-time over planned evaluation period of 1 year.

2.2 Data Processing Pipeline

The data is transferred through FTP on WiFi to the Windows computer. A telegraf service is running on the Windows machine constantly sending newly arrived accelerometer data to an influxDB instance that is deployed in the cloud. Fig. 2 shows a high-level architecture of the logging pipeline from the sensors to the cloud. The data is sampled during an acquisition window of 3 minutes at 1 kHz before it is sent to the Windows PC during a 17 seconds sampling pause for further handling (Fig. 3). Fig. 3 shows also the accelerometer signal at 3 axes as it is being acquired by the analogue digital converters (ADC) of the ESP32 WROOM board. We see that there are two dominant axes (adc0 and adc2) the third axis seems to pick up just noise. Similar data characteristics apply to the second vibration sensor. In addition to vibration data, manufacturing execution system (MES) data, as well as, other process data as product packaging speed is being collected in the same influxDB.

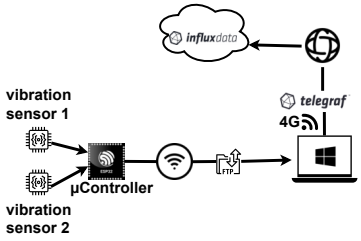


Figure 2: Overview of the data sampling pipeline from the vibration sensor into a time series data base in the cloud.

The process data allows for precise interpretation of the vibration data and is used for labelling in the later evaluation step. Here, we use python scripts to identify and prototype a suitable analytics and prediction framework with a goal to deploy it on site at the factory.

2.3 Data Quality Issues

The main data quality issue is related to data loss due to communication overhead and malfunction of the hardware. The microcontroller sampling routine as well as the process to send the data over FTP to the Windows server runs as a single thread. While the data is being sent to permanent storage, no data acquisition can take place in the current set-up. The duty cycle hence consists of 3 minutes data acquisition followed by a 17 seconds of data transfer as seen in Fig. 3, resulting in an 8.6% data loss per acquisition cycle. This strategy is sub-optimal in the current research setting where we are aiming to detect events on a sub-second scale. Further we experienced data loss of 1 month from one of the sensors due to cable wear from the mechanical abrasion of the connecting sensor cable.

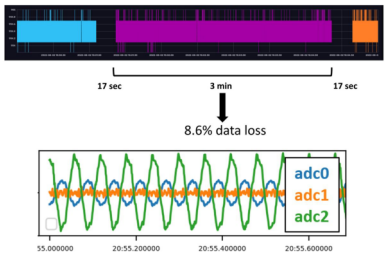


Figure 3: Top: Vibration data from a single accelerometer sampled at 1 kHz with acquisition gaps. Bottom: Vibration data from a single sensor consisting of 3 measurements at orthogonal axes.

3 SUMMARY AND OUTLOOK

We presented the acquisition pipeline consisting of the instrumentation set-up and software logging routines to be capable of collecting vibration data that is sampled at 1 kHz. The gaps in the data acquisition can be avoided by either making use of the multi-threading capabilities of the microcontroller or reducing the sampling rate such that the acquired data can be sent in very small chunks to minimise the communication overhead. The sieving unit vibrates with a frequency of 16 Hz, and we assume that the information that is relevant for the task of performance optimisation of the crushing facility is in the range of few hundred Hz therefore posing a lower requirement towards the sampling frequency. The experimental data acquisition set-up is meant to collect vibration data for one year. Once the data collection is finalised, we aim at building an analytics pipeline that allows us to correlate process data from the manufacturing execution system (MES) to optimise for key performance indicators downstream as for example the packing speed of the ferrosilicon in bags to be shipped to the customer. Further, we aim at being able to tackle questions regarding predictive maintenance of the facility by analysing extraordinary events detected through the vibration measurements of the sieving unit.

REFERENCES

[1] World Economic Forum. 2017. Digital Transformation Initiative, Mining and Metals Industry. In *White paper.*

Data Quality Issues in Solar Panels Installations: A Case Study*

Dumitru Roman
Antoine Pultier
SINTEF Digital
Oslo, Norway

Xiang Ma
SINTEF Industry
Oslo, Norway

Ahmet Soylu
Oslo Metropolitan University
Oslo, Norway

Alexander G. Ulyashin
SINTEF Industry
Oslo, Norway

ABSTRACT

Solar photovoltaics (PV) is becoming an important source of global electricity generation. Modern PV installations come with a variety of sensors attached to them for monitoring purposes (e.g., maintenance, prediction of electricity generation, etc.). Data collection (and implicitly the quality of data) from PV systems is becoming essential in this context. In this position paper, we introduce a modern PV mini power plant demo site setup for research purposes and discuss the data quality issues we encountered in operating the power plant.

CCS CONCEPTS

• **Information systems → Information systems applications**.

KEYWORDS

Solar panels, monitoring, data quality, data pipeline

ACM Reference Format:
Dumitru Roman, Antoine Pultier, Xiang Ma, Ahmet Soylu, and Alexander G. Ulyashin. 2022. Data Quality Issues in Solar Panels Installations: A Case Study. In *Proceedings of the 2nd International Workshop on Software Engineering and AI for Data Quality in Cyber-Physical Systems/Internet of Things (SEA4DQ '22), November 17, 2022, Singapore, Singapore.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3549037.3564120

1 SOLAR PANEL INSTALLATION AND MONITORING

As a major renewable energy source, solar photovoltaics (PV) [9] nowadays provide 3.1% of global electricity generation. PV monitoring is an essential part in any PV plant. Monitoring sensors and their working principles, controllers used in data acquisition systems, data transmission methods, and data storage and analysis technologies are very important in a monitoring system [3]. PV system monitoring may be the best way to maximize the performance of PV systems. However, each monitoring system affects in a different way the PV system performance [4]. PV monitoring systems have been proposed in the literature, e.g., based on open-source solutions with wireless and low-cost systems [5]. Others focus on the design and implementation of microcontroller based wireless PV modules [1]. Diagnostic techniques and algorithms were proposed to monitor photovoltaic plants, to predict failures and to enhance PV system performance [8]. Recognition Technologies (RT), Artificial Intelligence (AI), and Machine Learning (ML) enable drones and make the monitoring of large-scale solar power plants easier [2]. Data collection (and implicitly the quality of data) from PV systems is essential in this context, not only for better maintenance but also for better prediction of electricity generation.

A modern PV mini power plant demo site with 58 solar panels was installed on the roof of SINTEF building at Forskningsveien 1, Oslo, Norway, for research purposes, amongst others to collect and analyze the data from the PV plant and its associated sensors. Fig. 1.a shows a picture of the installation. To monitor the PV plant performance, various sensors are required. These include environmental sensors, data loggers, infrared cameras, etc. A CMP6 pyranometer (Fig. 1.b), a DustIQ soiling monitoring system (Fig. 1.c), and a ClimaVUE50 mini weather station (Fig. 1.d) are selected and installed for monitoring purposes. The Tigo system[1] is deployed to monitor the current, voltage and electricity output from each panel group through module optimizers and invertors. Thus, the information about radiation, dust related parameter, wind speed, environmental temperature, panel temperature, and power generation values can be monitored in real-time.

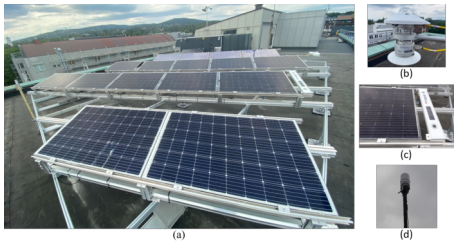


Figure 1: (a) PV demosite at SINTEF in Oslo, Norway; (b) CMP6 Pyranometer; (c) DustIQ Soiling Monitoring System; (d) Campbell Scientific CimaVUE 50 weather station.

Based on the large data that is collected, this demosite provides a unique opportunity to evaluate the power generation performance and explore the relation between different environmental variables which influences the energy output. For this purpose, a data pipeline was designed and implemented to collect and store data, and make it available for analysis. Fig. 2 depicts the data pipeline: data is collected from the sensors installed on the solar panels, as well as related sensors (e.g., from invertors, weather station), but also from external data providers (e.g., weather forecasting). Data that comes from proprietary systems (e.g., Tigo, SMA[2]) is firstly transmitted to corresponding proprietary cloud systems, after which it is downloaded, integrated/merged with the other data in the form of time

*This work received partial funding from the projects DataCloud (H2020 101016835), Super PV (H2020 792245), BigDataMine (NFR 309691), and SINTEF SEP-DataPipes.

SEA4DQ '22, November 17, 2022, Singapore, Singapore

ACM ISBN 978-1-4503-9459-8/22/11.
https://doi.org/10.1145/3549037.3564120

[1] https://www.tigoenergy.com
[2] https://www.sma.de

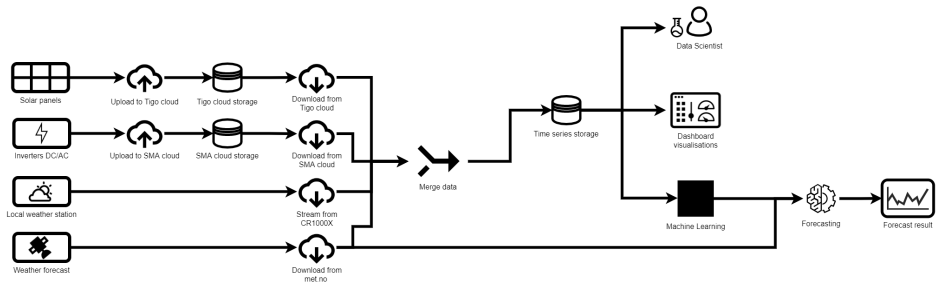


Figure 2: Data pipeline for collecting data from the PV installation.

series data, and stored for further analytics (ranging from basic visualization on dashboards to advanced AI/ML analytics). In the process of designing and implementing this pipeline we identified and experienced various data quality issues which we describe in the following section.

2 DATA QUALITY ISSUES

Missing Data. Missing data is a common problem that many reasons can cause. It can be hardware or software faults and last from seconds to weeks. We, for example, experienced the loss of electrical power for our data logger, which took a few days to fix. Our data is also streamed to the cloud using an IoT gateway and a broadband internet connection, which can be unreliable from time to time. The software could be missconfigured. For example, we kept only the last three months of data in the data logger at the beginning of the experiment, while we thought we kept everything forever. We also identified the risk of major hardware failures, such as broken cables or sensors. When data is missing for a very short period of a few sensors, interpolation can be considered. However, when the data is missing for longer periods, the data should not be used for analysis, especially for ML analytics. Removing the whole period is better than using default or dummy data.

Inconsistent Timing. The data sources in our systems have different sampling rates, from a few seconds to one hour, with clocks that are not necessarily synchronized tightly. We also experienced issues with time zones. Some sources also have delays in their availability. For example, a sensor uploads its data at a one-minute sampling rate, but only every half hour. We can use sensor fusion software methods to address such problems. We need to synchronize the timestamps, though it can be tricky to figure out the minor differences between the clocks. We also sometimes need to have a single sampling rate for all sensors, and we need to decide how we interpolate or sample the data. Average and linear interpolation are the primary solutions we use, but more advanced methods can also be used. For stream processing, we need to buffer the data for long periods, waiting for all our sensors to upload their data.

Unknown Condition. While an utterly broken sensor can be easy to spot, a sensor that produces inaccurate data can be challenging. Perhaps one temperature sensor fells from its solar panel to the floor while still reporting temperatures. One sensor may not be calibrated correctly or be replaced by another model with different characteristics. One classic solution in the big data domain is to use more data, so these issues are merely noise and could be ignored. In our case, we need to detect the quality changes using fault detection algorithms and careful data analysis.

Changes in the Experiment Environment. When doing an experiment outdoor for an extended period, years, for example, we should be prepared to observe significant changes in the environment. For example, we observed a new construction that obstructed the sun for a major part of the time for some of our solar panels. The panels could also be relocated or have their position adjusted. In our case, we decided to simulate the differences in sun exposure for the solar panels that are now mainly in the shade. But sometimes, the data should be dropped from the datasets.

Not Large Enough Experiment. We would like to have many years of data with many weather conditions in many locations, which would significantly increase the value of the data. One solution would be to share the data. People already share the energy production with little weather information on websites such as PVOutput[3]. Having a few more sensors in such community-sourced datasets would be valuable.

3 SUMMARY AND OUTLOOK

We introduced a modern PV mini power plant demo site and discussed the data quality issues we encountered in operating the plant. In future work we plan to identify and implement specific data pipeline solutions and strategies addressing the identified data quality issues [6, 7].

REFERENCES

[1] M Reyasudin Basir Khan et al. 2012. Wireless PV Module performance monitoring system. In *Proceedings National Graduate Conference 2012*. 1–4.
[2] Nallapaneni Manoj Kumar et al. 2018. On the technologies empowering drones for intelligent monitoring of solar photovoltaic power plants. *Procedia computer science* 133 (2018), 585–593.
[3] Siva Ramakrishna Madeti and SN Singh. 2017. Monitoring system for photovoltaic plants: A review. *Renewable and Sustainable Energy Reviews* 67 (2017), 1180–1207.
[4] Eneko Ortega et al. 2017. Study of photovoltaic systems monitoring methods. In *2017 IEEE 44th Photovoltaic Specialist Conference (PVSC)*. IEEE, 643–647.
[5] José Miguel Paredes-Parra et al. 2018. PV module monitoring system based on low-cost solutions: Wireless raspberry application and assessment. *Energies* 11, 11 (2018), 3051.
[6] Dumitru Roman et al. 2021. Big Data Pipelines on the Computing Continuum: Ecosystem and Use Cases Overview. In *Proceedings of the Symposium on Computers and Communications, 2021*. IEEE, 1–4.
[7] Ahmet Soylu et al. 2022. Data Quality Barriers for Transparency in Public Procurement. *Inf.* 13, 2 (2022), 99.
[8] Asma Triki-Lahiani et al. 2018. Fault detection and monitoring systems for photovoltaic installations: A review. *Renewable and Sustainable Energy Reviews* 82 (2018), 2680–2692.
[9] Marta Victoria et al. 2021. Solar photovoltaics is ready to power a sustainable future. *Joule* 5, 5 (2021), 1041–1056.

[3] https://pvoutput.org

Author Index