Aalto University

OPEN ACCESS

This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Truong, Linh; Nhu Trang, Nguyen Ngoc

TENSAI - Practical and Responsible Observability for Data Quality-aware Large-scale Analytics

Submitted: 01/01/2022

Published under the following license:
CC BY

Please cite the original version:
Truong, L., & Nhu Trang, N. N. (2022). *TENSAI - Practical and Responsible Observability for Data Quality-aware Large-scale Analytics*.

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

TENSAI – Practical and Responsible Observability for Data Quality-aware Large-scale Analytics*

HONG-LINH TRUONG†, Department of Computer Science, Aalto University, Finland
NGUYEN NGOC NHU TRANG‡, Daienso Lab, Vietnam

Given a large-scale mobile network with a variety of equipment and radio access networks technologies for an approximate 20 million subscribers, there are many types of data that can be used for big data analytics and machine learning (ML) tasks for network operations, monitoring, and optimization. However, a variety of data is measured, collected, and propagated through numerous complex data and software systems. Thus, people, software components, and data-driven operations for big data and ML pipelines face great challenges in dealing with data quality impacts. Data quality related problems occur and are propagated through complex operations involving different types of data, people, software components, and analytics that cannot be solved purely through data quality engineering. This paper discusses our TENSAI framework, as practical and responsible observability for ensuring data quality in such a mobile network. TENSAI focuses on methods of communications, strategy specifications, and data quality engineering for diverse types of data and analytics among different types of operations. TENSAI presents techniques for capturing and communicating causes/effects about data quality problems clear to all relevant stakeholders, developing data quality-aware adaptation strategies for actions on data that can be integrated into analytics processes, and engineering the data quality awareness in software and data pipelines. Thus, TENSAI supports full visibility of data quality problems and impacts among related systems to empower the utilization and adaptation of data analytics for different types of operations. We will illustrate our TENSAI with several real-world data types, pipelines, and cases based on our mobile network.

Additional Key Words and Phrases: data analysis, machine learning, data quality, telecommunication networks

1 INTRODUCTION

We are working on various big data and machine learning (ML) pipelines for diverse types of operations in a large-scale mobile network. Our mobile network consists of a variety of radio access networks (called "V-RAN" in this paper), which include 2G, 3G, 4G, and 5G (denoted as "2-5G") radio access networks from different hardware and software vendors (such as Nokia, Ericsson and Huawei) in Vietnam. There exist many (legacy) complex systems to capture different types of data. From these systems, different types of batch and realtime analytics, including ML-based predictions, are being developed for customer service issue resolution, traffic prediction, subscriber's quality of experience, site alarm anomaly detection, equipment's predictive maintenance, to name just a few. Such analytics must obtain various types of data for different big data analytics and ML pipelines carried out through a set of complex software. They involve different types of data extraction and (pre-)processing functions to provide data for analytics. Thus, they must support quality data delivery through complex pipelines. However, such systems, analytics and their complex software are strongly affected by the operations of the V-RAN hardware and infrastructures managed by network system operators. Generally, we can see *three main types of operations* involved in data-driven monitoring and optimization of V-RAN: (i) the operations of V-RAN infrastructure's runtime monitoring, deployment and maintenance – called *V-RAN operations*, (ii) the operations of data engineering of software/data components and pipelines – called *data engineering operations*, and (iii) the operations of suitable data analytics and ML algorithms/services – called *analytics operations*.

*(c) 2022 Copyright held by the owner/author(s). First version released on 25.08.2022. Minor update of the author affiliation and paper format on 14.06.2023.
†Correspondence email: linh.truong@aalto.fi
‡Work carried out when the author was with Mobifone Corporation, Vietnam. Correspondence email: nhutrang.nguyen@daienso.com

The above-mentioned three types of operations are carried out across systems of data and software components with different teams of network, software, data and ML engineers and scientists. Data quality problems are propagated through complex software pipelines (*data engineering operations*) to analytics/ML (*analytics operations*) may be caused by changes in V-RAN infrastructures (*V-RAN operations*). Thus, to solve the problems, we must have a holistic view on how data is collected, preprocessed, and moved through complex pipelines into the final batch and realtime analytics for decision making. Currently, network system operators, software engineers, and data scientists/engineers do not have a whole picture of data quality impacts and possible actions related to their operations. Their interactions and communications centered around data-driven operations are asynchronous, spanning in different times and spaces. They may realize data problems too late or they are not aware of the problems, which cost them effort and slowdown their operations. Thus, it is of paramount importance to establish a framework for enabling a holistic view and synchronizing the data quality problems and impacts on different types of operations.

To date, the major works in data quality assurance in big data analytics for such a large-scale network are either for a specific type of data or a specific data pipeline, without a traceability and communication of sources of data quality problems and impacts. Furthermore, the discussion in these works is focused on the data itself, but not on other important aspects in terms of communications, strategies, software engineering, and involved teams centered around how to act appropriately in data pipelines and data products. In many cases, existing solutions are just focused on detecting and reporting data quality for data observability [28] or on solving data quality at a certain point in the data pipeline and within ML algorithms [7, 11, 17, 37]. These works are important, however, they are not enough as the problem of data quality impacts cannot be solved by and in algorithms alone in a complex network like V-RAN, of which operations require intensive domain knowledge associated with operations and business contexts. In typical data science processes, different roles just concentrate on the data passing through data collection, processing, and training [35] without understanding changes in the systems generating the data (hardware, software, and subscriber infrastructures). In large-scale systems, data quality problems must be examined with root causes from these infrastructures. Furthermore, reactions to data quality problems are strongly based on operation and business and geographical contexts. Hence providing data quality measurements alone is not enough. In order to understand what problems might occur, we must capture, communicate and provide detailed possible sources of problems and implement addons to support the operators and developer.

In this paper we contribute TENSAI (pracTical and rEsponsible eNd-to-end obServability for dAta qualIty) as a framework. We present TENSAI as a set of methods and services for identifying key data quality problems, communicating the problems, defining strategies, and engineering solutions to tackle the problems. TENSAI suggests solving the data quality problems from three perspectives: (i) making and communicating cause/effect information clear to all relevant stakeholders, (ii) developing contractual strategies for actions in cases of data problems that can be integrated into the software development/operations processes, and (iii) engineering the data quality awareness in data pipelines. TENSAI is devised to ensure that solving data quality impacts follows the continuous involvement of multiple stakeholders across different pipelines and at different times. For managing V-RAN, TENSAI also provides methods to help improve disparate management in choosing and executing data analytics for operations by enhancing communications and insights for operators to use valuable data analytics. To date, data quality and impacts in V-RAN has not been effectively governed (monitoring and controlling) to enable adaptation and change management on the system due to the lack of coherent communication, strategies, and engineering. TENSAI helps to remove difficulties in handling situations arising in operations and to ensure accuracy and reliability for ML tasks, allowing data quality to be checked regularly and

throughout data-driven operations. This paper also contributes several real use cases to illustrate TENSAI's usefulness in an industrial setting.

The rest of this paper is organized as follows: Section 2 presents relevant data. Section 3 presents key elements of our TENSAI method. Section 4 presents main cases. Further related work is discussed in Section 5. We conclude the paper and outline our future work in Section 6.

2 DATA, PIPELINES AND ANALYTICS CASES

2.1 Data Types

Data types/-data sources	**Data format**	**Atomic Level**	**Measurement type**	**Example of data quality problems/-consequences**
Customer feedback from subscribers and call centers	time series records, including both structured data and free text inputted by humans	single feedback from a single subscriber	geographical location, serving cell/site, complaints, cause, datetime	data currency when handling customer feedback, inaccurate data due to text processing
Network measurements (NMs) from Operations Support Systems (OSS)	time series records, each capturing a set of measurements	cell, site and zone (district, province and user-defined zone)	availability, accessibility, mobility, throughput, latency, traffic (voice, data), utilization (transmission, radio)	missing data due to recording and data capture, incorrect data due to a software update, changed counters, or infrastructure changes
Alarms from Alert Management	time series records, text-based machine generated logs, and annotations by operators	equipment, power and battery system, quality of service, communications	severity, fault id, hardware unit name	incorrect or missing data due to text processing, change of the data details
Incidents from humans/operator or incident monitoring	time series records, structured data or free text inputted by humans	cell, site, Base Station Controller (BSC), Radio Network Controller (RNC) and zone (district, province and user-defined zone)	severity, affected network measurements/services, cause	incorrect data due to text processing or missing data due to non-pattern text records, name identity change

Table 1. Types of data and their possible data quality problems. The data is from a V-RAN of 2-5G radio access networks with equipment and devices from different vendors.

Table 1 shows the four major types of data that we consider in our V-RAN. *Customer feedback* captures data related to subscribers and their feedback about the usage of the mobile network. *Network measurements* capture hundreds of different types of counters and many high-level network performance and traffic indicators. *Alarm data* captures many infrastructures and hardware alarms as well as high-level operational alarms. *Incident data* captures key incidents in V-RAN operations. In each type of data, we have many different details, due to the existing of different technologies in radio access networks (2-5G), equipment, software vendors, and network subscribers.

For the purpose of analytics and due to the deployment technologies, these types of data are stored using different technologies and extracted/integrated into different data lakes (structured and semi structured data) for analytics using current technologies (e.g., file/object storage with Minio[1] and data lakes with Apache Hudi[2]). These types of data can be used to create different data resources used for different analytics purposes. Conceptually, each *data resource* can be represented as a *dataset*, under a limited size or in a continuous stream. Dataset can contain only selected elements of a type or a mixed set of elements from different types in Table 1. Often a dataset is read-only or processed with the insertion of new data. Therefore, any quality of data detected might happen in the data analytics and the detection of data problems might not be used to correct the (historical) data in the data lakes/storage. Although we discuss with data in V-RAN, similar situations may be found in other complex infrastructures and systems where one must operate multiple systems with diverse technologies to serve a large number of consumers.

2.2 Complex Software Systems and Data Pipelines

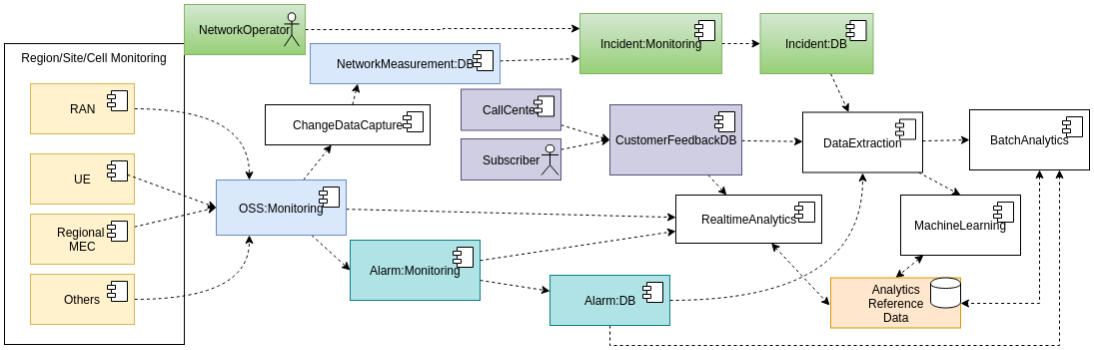

Fig. 1. A set of systems for different data pipelines. Similar colored components are in the same system for monitoring and collecting data, while the rest is related data analytics/ML ones. Originally the source of data is from approximately 70000 sites/cells across four mobile network technologies (2G, 3G, 4G, and 5G) and millions of subscribers. Along the pipelines, data quality problems may occur in different components and propagated cross systems. Apart from machine errors, human operators and subscribers can introduce many data problems.

Our V-RAN has different, complex software systems based on that various types of data and analytics/ML pipelines are processed and executed. Data quality problems and corresponding actions must be identified, analyzed and understood along these pipelines. Figure 1 provides a high-level view of software systems and selected, important pipelines for data mentioned in Table 1. Due to the complexity of these software and business sensitive deployment information, we only provide a high-level view of original sources of data and four groups of systems:

- Regional, site, and cell monitoring (`Regional/Site/Cell Monitoring`) captures various types of measurements across the whole infrastructure. They are one of the original sources of data for analytics. Such measurements enter into a complex OSS (`OSS:Monitoring`) and eventually to databases/storage (`NetworkMeasurement:DB`) for analytics. OSS monitoring and corresponding databases establish the first system – the Network Measurement (NM) system.

[1]https://min.io/
[2]https://hudi.apache.org/

- Alarm monitoring (`Alarm:Monitoring`), which is based on OSS data, will provide alarm data stored into alarm database (`Alarm:DB`). Together they establish the second system – the Alarm system.
- Incident monitoring (`Incident:Monitoring`) is based on network measurement and network operators (human operators, `NetworkOperator`). Incident data is stored into a database (`Incident:DB`). All together they introduce another original source of data and the third system – the Incident system.
- Subscriber (`Subscriber`) and call center (`CallCenter`) gather customer feedback which are stored into a database (`CustomerFeedback:DB`). Together they introduce another original source of data and the fourth system – the CustomerFeedback system.

In our work, these systems are available based on different software and compute infrastructure technologies. Operating and monitoring V-RAN infrastructures `Region/Site/Cell monitoring` and `OSS:Monitoring` is within the *V-RAN operations*, data components and pipelines are under *data engineering operations*, and analytics is under `analytics operations`. We focus on pipelines extracting and analyzing data from these systems for various tasks (realtime analytics, ML, etc). Such pipelines are within and across components like `ChangeDataCapture, DataExtraction, BatchAnalytics, RealtimeAnalytics` and `MachineLearning`[3]. These components are complex and built atop various state-of-the-art software like Apache Spark, Apache Flink, Apache Kafka, Apache Hudi, TensorFlow, and data science programming toolkits. `Subscriber, NetworkOperator, CallCenter` are components with human inputs. Thus, they produce a higher rate of quality problems. Realtime analytics, batch analytics and ML inferences are relied on various types of `Analytics Reference Data`, which capture domain-specific thresholds, patterns, and profiling data based on operations and business contexts. Results from analytics and ML are also extracted and studied for improving `Analytics Reference Data`.

2.3 Operation Use Cases

With the types of data in Table 1 in our large-scale data analytics in Figure 1, there are many different operation areas, which require different data resources, analytics, and specific V-RAN, data engineering and analytics operations. Table 2 shows key operation areas. There are a large number of analytics and ML methods that can be used for analyzing data to support operations in these areas. Such analytics and ML methods have different capabilities in terms of service performance (fast or slow), analytics accuracy (quality of results and ML models), and abilities to work with problematic data (missing data).

In utilizing these analytics for such operations, different analytics techniques, including ML, must be used in the right way for suitable datasets under appropriate data quality conditions. In addition to data sources for analytics/ML, business strategies, service/customer stratification and priorities, and geographic service zoning (e.g., dense urban, urban, and rural region) are also important aspects in domain analysis/ML for operators to make decisions. Therefore, the couplings among data, operation and business contexts, and algorithms are key information for us to develop suitable actions given detected data quality problems.

3 TENSAI FRAMEWORK

To capture cause/effect of data quality problems for complex systems shown in Figure 1, all relevant stakeholders (such as, network system operators, data quality manager, and data scientist) must be connected and must communicate and synchronize their view on data flows among software components and data products (including raw/preprocessing data

[3] Although `BatchAnalytics` and `RealtimeAnalytics` can use and implement ML methods, in this work we have `MachineLearning` as a building block to indicate specifically analytics utilizing ML methods. `MachineLearning` can be carried out in batch or realtime modes.

Operation areas	Description	Data	Analytics
Response & Recovery Plan	Optimizing response and recovery tasks in cases of natural disasters or blackouts (or similar situations)	Alarm, Network Measurement, Incident, Analytics Reference Data (Network coverage), Operator resources and their context (location and tasks)	Clustering, AI/ML planning
Traffic Understanding	Assessing and determining causes related to uploading/-downloading traffic	Subscriber, Network Measurement, Analytics Reference Data (Cell/Site group profiles)	Forecasting, anomaly detection, classification, pattern similarity search and discovery
Feedback Serving	Automating feedback answering workflows for networks and networks information access for customer services	Alarm, Incident, Network Measurement, Customer feedback, Analytics reference data (Cell/Site/Zone profile), Training data (integrated data)	Real time and batch analytics, classification
TWAMP Transmission Effect	Evaluating customer feedback related to TWAMP (Two-Way Active Measurement Protocol) and the quality of transmission	Alarm, Network Measurement Key Performance Indicator (KPI), latency, packet loss, customer feedback	Classification, causal inference
Site performance & Customer feedback	Evaluating site performance effects on customer feedback	Performance indicators from Network Measurement (e.g., Key Quality Indicator (KQI), Physical Resource Block utilization, KPI, packet loss, congestion rate), Alarm, customer feedback	Classification, prediction, anomaly detection
Electricity usage & Operation cost	Analyzing causes for electricity usage and evaluating costs and anomaly operation costs of Cell/Site	Power profiles, electricity consumption, electricity bill Correlated data among various sources	Anomaly detection, forecasting, classification
Network Usage & Performance Forecasting	Forecasting data usage and transmission loads	Historical usage data, subscriber, Network measurement,	Forecasting, anomaly detection, clustering
What-if Blackout/Emergency Situation Simulation	Assessing impact of blackout to network infrastructure and evaluating possible blackout of the network	Network Measurement, Incident, Power/Electricity data, Analytics reference data (Equipment, cell and site profiles)	Forecasting, causal inference, recommendations
Customer Churn Analytics	Analyzing causes for customer retention and forecasting data usage	Subscriber, Customer feedback, Network Measurement, Incident	Forecasting, classification

Table 2. Examples of operation areas, data and analytics

and analytics results). Such connections and communications, via observability techniques and enabling observability services, will provide a global picture of changes, data quality impacts, etc., in an end-to-end manner that will be implemented via a set of services and processes. From their view on a pipeline, they can see data quality changes/problems propagation through the pipeline. Based on that, suitable actions can be performed. The approach supports three aspects: communication, contractual data quality, and quality-aware engineering, all together.

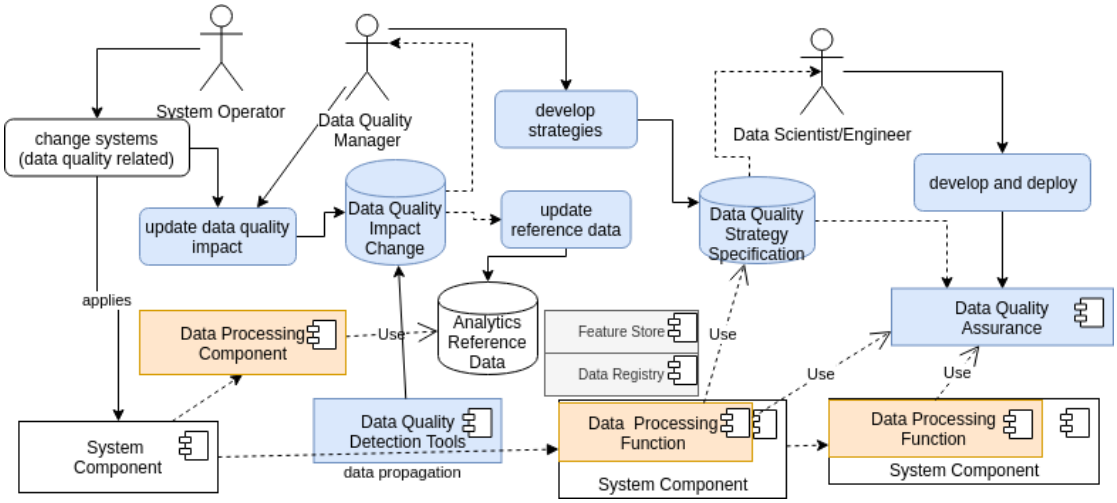

Fig. 2. TENSAI activities, components, and interactions: `System Components` represent services/platforms with V-RAN whereas `Data Processing Functions/Components` illustrate functionality used to collect, extract, ingest and process data that can be separated or embedded within `System Components`.

Figure 2 outlines the TENSAI framework centered around the following key points:

- The identification of roles in different types of operations must be clear and these roles must communicate w.r.t. data quality problems and impacts in a proactive and/or reactive manner. Three categories of roles are `System Operator` – working on system/infrastructure change and deployment, `Data Quality Manager` – overseeing the data quality governance processes and appropriate strategies, and `Data Scientist/Engineer` – working on data analytics algorithms including ML and data pipelines. These roles spread across different divisions and work across software and data systems in V-RAN, but they are linked via dependencies of data and corresponding analytics in V-RAN. Note that a person can play the role of `System Operator` when changing configurations/parameters in V-RAN, and the same person can also act as `Data Quality Manager` to report change impacts for analytics. A role can be carried out by an individual or a team.
- Changes of V-RAN systems and infrastructures, during *V-RAN operations* affecting the quality of data used by other operations must be updated into `Data Quality Impact Change` (e.g., which types of systems have been changed). Data quality impact changes, see Figure 3, must be taken for the development of `Strategy Specifications`, e.g., by `Data Quality Manager`, in collaboration with `Data Engineer/Scientist`, to deal with the impact changes, e.g. in *data engineering/analytics operations*. Strategies are based on data quality metrics but defined for specific business and operations contexts, reflecting via constraints on quality and possible actions. All related `Data Processing Component` must be linked to strategy specifications. This link can be reflected via software implementation or documents to make sure that analytics and people work on the pipeline know the strategy specifications, e.g., during `analytics operations`. Furthermore, any `Analytics Reference Data` used for `Data Processing Components` must be updated if the quality impact change affects the reference data. Especially, the reference data is used to support *data engineering operations* and *analytics operations*, such as group conditions, anomaly detection thresholds for specific service type and service area, and site capacity due to various hardware configurations.

- Some strategy specifications must be implemented into `Data Quality Assurance` which will be integrated into suitable data and software components for automating data quality control and data quality-aware processing. `Data Quality Detection Tools` must be provided to automatically capture and report possible changes and data quality problems.

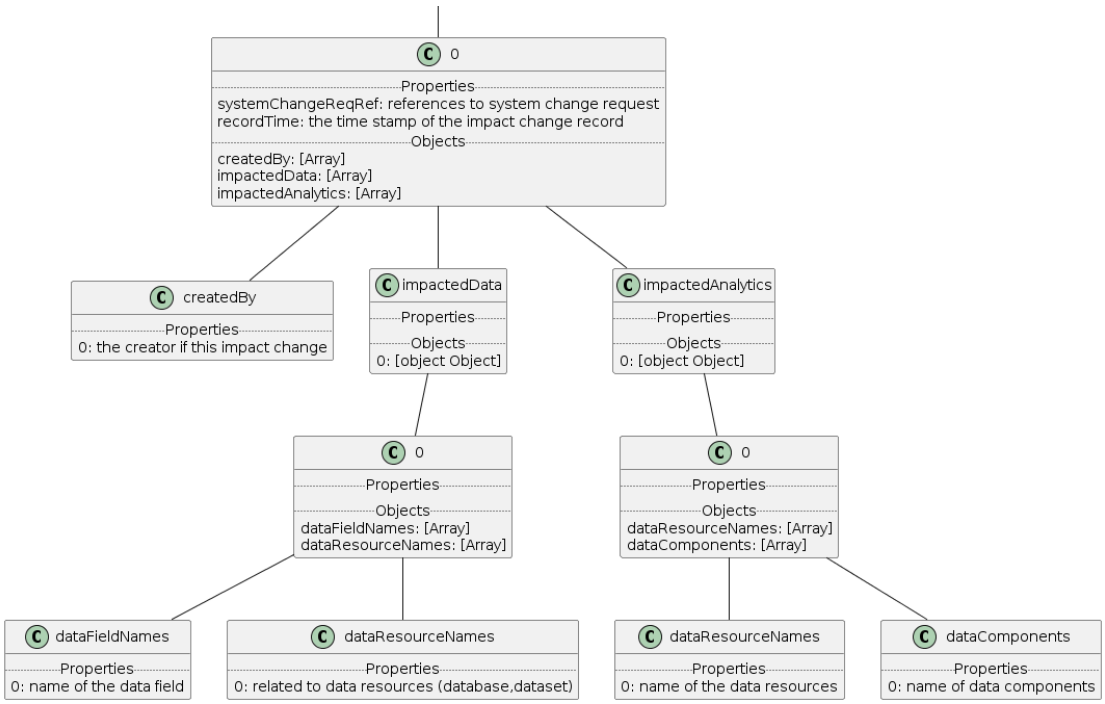

Fig. 3. Data quality impact change: two main types of information are impacted data (data fields and data resources) and impacted analytics (batch, realtime and ML analytics in existing data components and tools)

Essentially, TENSAI deals with (i) change management w.r.t. data quality impact and its association with analytics/operations for different contexts, (ii) automatic data quality detection and strategy specifications for actions based on data quality, and (iii) software components for data quality aware processing. These aspects happen in a cross-operation, cross-team and cross-system manner that require novel ways to communicate, coordinate and manage actions based on data quality observability for appropriate business and operation contexts:

Cause/effect identification and communication: One of the first cross issues is to capture and manage system changes impacting data quality. For example, V-RAN system configurations and upgrades often lead to changes in network measurements. Given the changes, an evaluation can be executed to produce initial assessment of changes that `Data Quality Manager` can revise and update into `Data Quality Impact Change`. System operators carrying out configuration changes (for network operation or service optimization purposes) will not know all possible impacts on data quality because the operators do not have a big picture of possible data usage in many different data pipelines and potential impacts. Therefore, capturing change impact requires the communication between `System Operator` and `Data Quality Manager`. In some cases, `Data Quality Manager` will evaluate the change and update the impact alone.

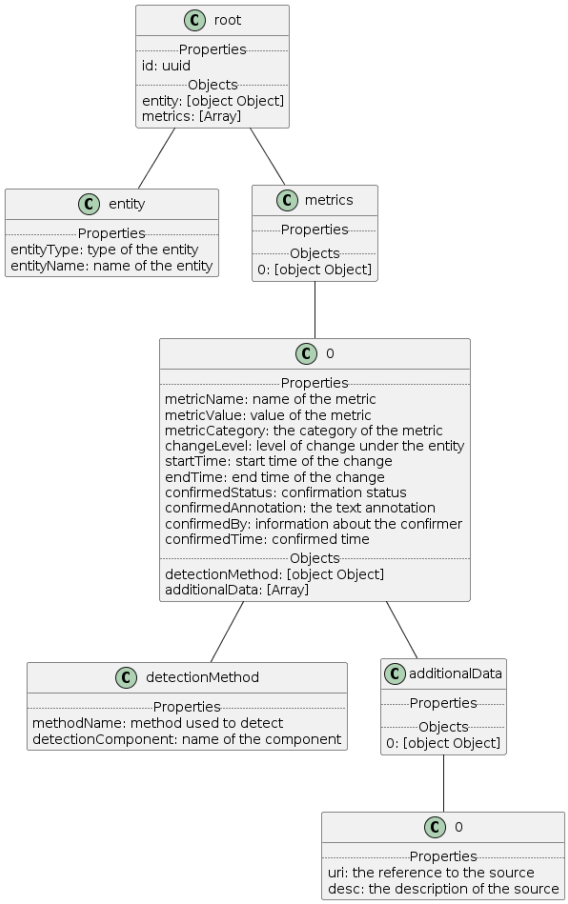

Fig. 4. TENSAI communication messages

Furthermore, `Data Quality Manager` may not be aware of changes or does not see all potential impacts. Therefore, `Data Quality Manager` might receive feedback from automatic `Data Quality Detection Tools`, which monitor data flows and update information about impacts. Such detection tools are needed for large-scale systems and we can have different ways to implement and integrate them into the systems, such as explained in [46]. In advanced situations, different ML techniques and human-in-the-loop can be combined to build an advanced detection tool of changes [2, 18].

Communications among teams and operations will be done via change notifications based on information stored in `Data Quality Impact Change` and `Data Quality Strategy Specifications`. In terms of capturing and documenting data quality impact changes and strategies, we rely also on existing `Data Registry` – which manages metadata about data resources and `Feature Store` – which manages data (features) used for training ML models[4]. These components are very common nowadays for managing big data resources and supporting ML pipeline engineering. They provide metadata about existing data resources, data schemas and features/data fields used for ML that we can link to the impact changes and strategies. The `Data Quality Manager` and `Data Scientist/Engineer` can work with these components

[4]https://www.featurestore.org/

to retrieve related data resources given a data field change. Naturally, we could configure the propagation of these changes through change management services for complex team operations (such as, PagerDuty[5] or Opsgenie[6]). In this case, changes can be propagated through communication services like Slack and Microsoft Teams[7]. Then a workflow can pickup to reconfigure the assurance component. Figure 4 presents TENSAI communication messages.

Strategies for specifying data quality contracts: Strategy specifications for data quality impact will govern possible actions given the assessment of data quality. In TENSAI, such strategy specifications can be turned into contracts that either (i) we can implement the contracts into software components for data processing or (ii) the stakeholders involved in relevant operations associated with the data will have an understanding and responsibility to follow the specifications. In the first case, the contractual terms can leverage those in data contracts used for data exchange among different parties [26, 45]. The core technical matter is that such contracts express data quality metrics, constraints and possible consequences to guide other data processing and analytics tasks. However, contracts cannot be specified using a single specification due to the diversity of data and possible actions. To develop them, we leverage key data quality metrics [3, 25, 36, 42] and metadata terms from existing `Data Registry` and `Feature Store`, such as Linkedin DataHub[8], Google Data Catalog[9], and Apache Atlas[10] and from data and service contracts, such as QoA4ML [47] for detailed, individual specifications and focus on managing them, using a model shown in Figure 5.

Quality-aware data pipeline engineering: Based on constraints in strategy specifications for data quality impact, `Data Scientist/Engineer`, who operates and/or develops a data pipeline or an analytics, must incorporate suitable `Data Quality Assurance` into the pipeline/analytics. Essentially, `Data Quality Assurance` in our framework represents and abstracts various different, concrete techniques, implemented in suitable software components, to evaluate data quality according to the strategy and to control related tasks as a consequence of the data quality evaluation. This means to utilize different tools and libraries, such as Python-deeque[11] and Great Expectation[12], and mechanisms, such as [46], to evaluate data quality and provide the quality assessment results to suitable components. Due to the diversity of data and pipelines, suitable methods must be implemented based on the type of data and the pipeline handling the data. There will be no single way, as in V-RAN (and similar to many large-scale systems) we have used different technologies for data engineering and analytics, such as streaming analytics with Apache Spark and Apache Flink, batch analytics with Apache Airflow, Apache Spark and Pandas, and ML with Tensorflows, Apache Spark, and scikit-learn. New components may be needed when a new strategy specification or problem emerges. One aspect is to reconfigure `Data Quality Assurance` if possible. This happens when some changes do not lead to the engineering and deployment of a new type `Data Quality Assurance` and it would be enough for a reconfiguration (e.g., format of data field or completeness of new data fields).

Figure 6 summarizes TENSAI technical features (the upper part). They all can be used for various use cases, such as the above-mentioned areas and use cases (the middle part) carried out by different operations in V-RAN (the lower part). In this work, we apply TENSAI for V-RAN but TENSAI is a generic framework. The key goals that TENSAI helps to achieve are:

[5]https://www.pagerduty.com/
[6]https://www.atlassian.com/software/opsgenie
[7]https://slack.com/ & https://www.microsoft.com/en/microsoft-teams
[8]https://github.com/datahub-project/datahub
[9]https://cloud.google.com/data-catalog/docs/concepts/resource-project
[10]https://atlas.apache.org/
[11]https://github.com/awslabs/python-deequ
[12]https://greatexpectations.io/

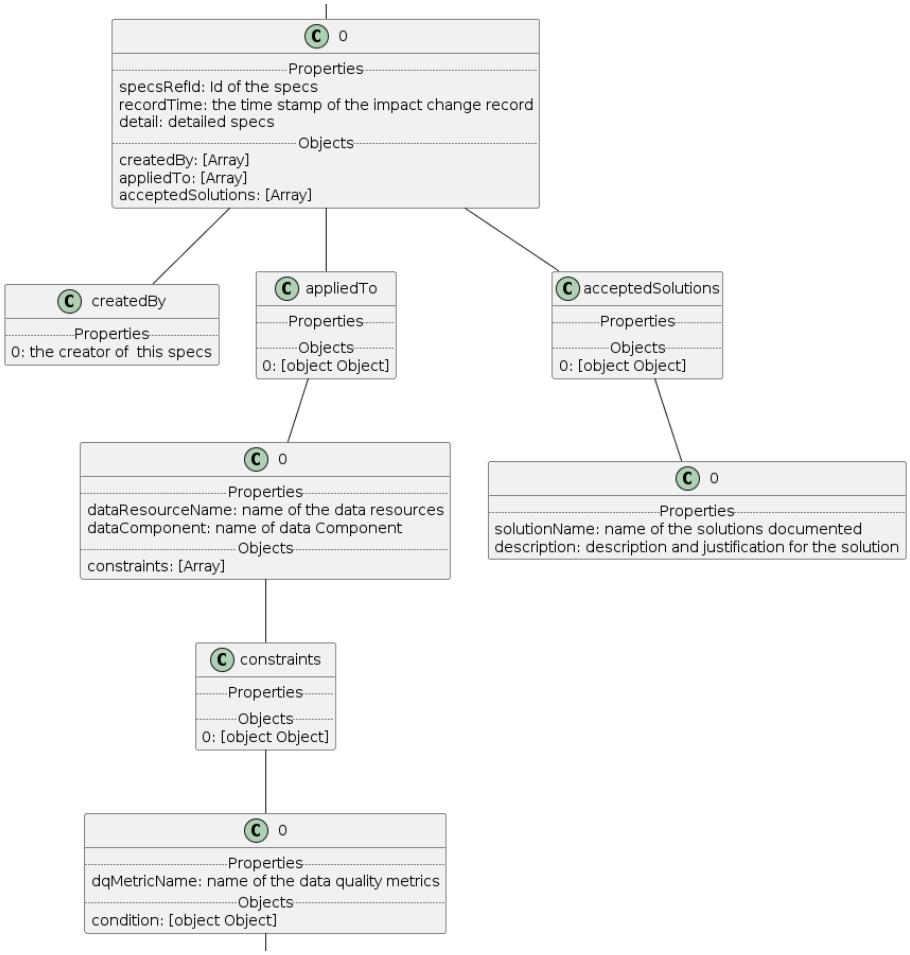

Fig. 5. Strategy specifications: mainly a strategy has (i) constraints of metrics for data resources and data components, and (ii)accepted solutions about possible actions, such as executing an analytics or delegating a problem to another operation.

- Manage and build quality-aware data pipelines that create and maintain expected data quality from the beginning of the cross-system.
- Implement suitable approaches/tools to capture changes from *V-RAN operations* along the cross-system, where are the main sources of data quality issues, reducing time and cost for operations in tracing root causes.
- Document quality impact and implement data constraints and data quality traceability for data pipelines.
- Provide an effective communication solution between teams (V-RAN operations, data quality managers, and data analysts/engineering) to react to data quality issues.

In the next section, we contribute several real-world cases supported by TENSAI.

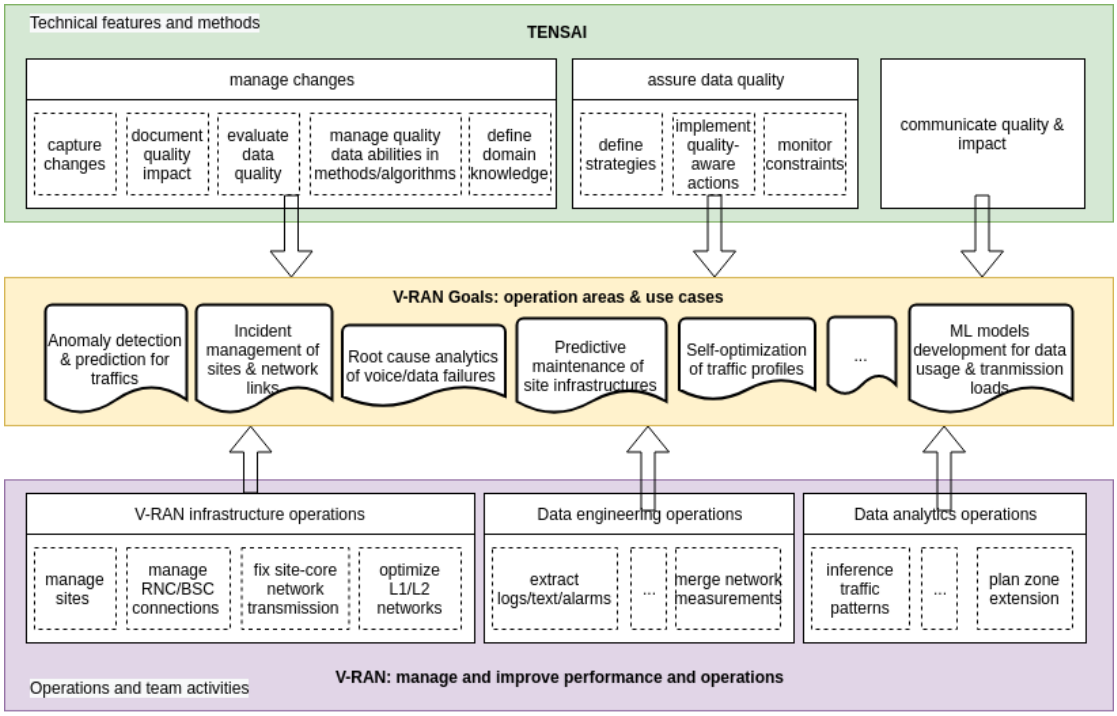

Fig. 6. TENSAI technical features for supporting expected goals/use cases carried out in different V-RAN operations

4 SOLVING PRACTICAL PROBLEMS WITH TENSAI

In this section, we will present a few key data quality problems across various components in our V-RAN (mentioned in Table 2 and Figure 1). Three problems, (i) entity changes, (ii) currency of data and its structure, and (iii) missing data, will be discussed. Specific cases and examples, related to V-RAN operations and analytics areas mentioned in Table 2, within these problems will be elaborated. For each problem, we will discuss how we apply TENSAI for solving the problem from three perspectives:

- *Cause/effect identification and communication*: capture changes and communicate problems,
- *Strategies for specifying data quality contracts*: define constraints on quality, and
- *Quality-aware data pipeline engineering*: possible engineering solutions for data and analytics pipelines.

The cases and examples will be discussed with our real data, code and designs[13].

4.1 Entity changes

4.1.1 Problem description. Due to a continuous deployment of cells/sites (or replacement) in the infrastructure of V-RAN, existing entities will be modified and new entities will be introduced (within *V-RAN operations*). This happens often with mobile sites/cells as the core entities in the V-RAN infrastructure. Therefore, changing the identity of an entity, such as changing the name and the location (physical GPS or logical tracking area code/location area code address)

[13]Due to the business sensitiveness of the data, code and designs, we will anonymize certain parts of data, provide simplified code and abstracted designs. Also some concrete parameters in use cases are not the real values V-RAN business and operations.

of sites/cells, is a common task. Another change is to deploy new sites/cells to improve network coverage and capacity (the evolution of the V-RAN infrastructure). Such changes lead to entity identity change in measurements collected by `Region/Site/Cell Monitoring` (see Figure 1). These changes are common tasks from the network management viewpoint, carried out by `System Operator`, but they have a strong consequence on data engineering and analytics, including `Realtime Analytics, BatchAnalytics` and `MachineLearning`, due to the lack of communication among stakeholders and of efficient management of the observability of data quality change through a complex chain of software components and data/ML pipelines. The problems of data drift, concept drift and schema drift can happen due to such changes. But the detection may not be seen immediately, e.g., in case of `BatchAnalytics` and `MachineLearning`. If not undetected, for example, in terms of new sites/cells, analytics may perceive a data completeness problem (at the time of analytics some data is seen as missing). However, actually it is not a data completeness problem. Instead, new data comes because of new entities. Furthermore, this issue has a strong impact on the type of analytics chosen and parameters for the choosing analytics, especially for algorithms for change point detection and prediction. The key reason is the drift in data, according to the domain knowledge. For example, in V-RAN, when new sites/cells are introduced into a zone, the new sites/cells will start taking the load in the zone and they interact with other existing sites/cells, such as via handover processes. Therefore, analytics related to network measurements and subscriber usages will be different, unlike the analytics for a stable mobile network zone.

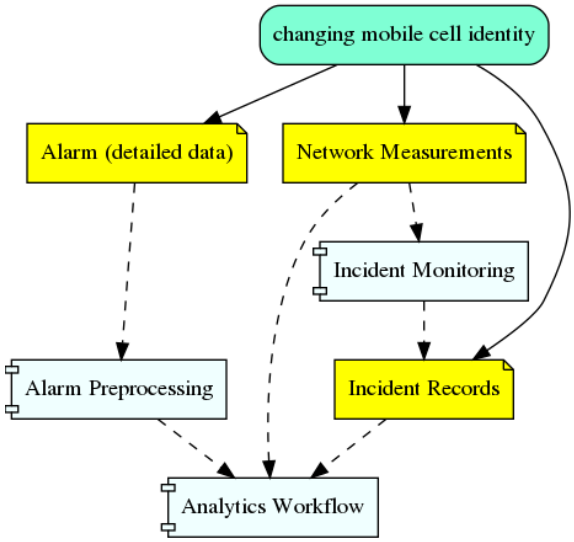

Fig. 7. Example of a cause-effect graph due to the change of identity for mobile cells as entities in our V-RAN

4.1.2 *Dealing with data quality impact using TENSAI.*

Cause/effect identification and communication: The effect of change on data quality impact can be captured clearly by the role of `Data Quality Manager` for main types of data. This must be done and enforced to provide insights for relevant stakeholders. Figure 7 shows the effect propagation of name identity changing, as one important observed insight for communicating possible data problems in V-RAN. This effect is expressed in terms of a cause-effect graph

with detailed information about possible data fields and data resources to be impacted. Then we use the graph to communicate to different stakeholders and product owners of data analytics.

In terms of management of changes, once we apply an identity change of an entity, subsequent network measurement, alarm, and incident data associated with the entity will be with the new identity. The change management system will capture different information, such as start time, changing entity, trend, and change patterns. Captured information about changes is documented as records and stored into a centralized place in TENSAI framework. Following TENSAI, the source of V-RAN system change is linked to the data quality impact change record stored in (`Data Quality Impact Change`). The list of affected analytics will be associated with types of data resources linked to data quality impact change records. Thus, any analytics developer and owner will be aware of potential effects due to the change. For example, Listing 1 is the original change configuration for system components w.r.t. cell/site change, whereas Listing 2 provides a record for data quality impact change. Note that the documentation process might not be fulfilled in practice. Therefore, the above-mentioned way may not be suitable for certain types of local or automatic changes. When the problem is unknown to `System Operator`, the change information cannot be documented by humans (operators, managers) into the change management system. Therefore, we also detect changes by using external data evaluation tools for change point detection and anomaly detection [16, 34, 41]. Captured information by these tools will be updated into `Data Quality Impact Change` and a human-in-the-loop approach is used to inform operators, managers, or scientist/engineers about the situation.

```
{
"ChangeRequest_Number": "CR_RAN_LTE_20220512_090924.208",
"ArtifactLink":"URI...",
"CR_Name": "Change_cell_name",
"Status": "Waiting",
"SDATE": "12/05/2022 09:09",
"EDATE": "25/05/2022 09:09",
"EXECUTED_DATE":,
"RISK": "NO",
"Service_affected": "NO",
"Province": "***"
}
```

Listing 1. An example of a change record for renaming a CELL. Note that *** is used to mask sensitive data

```
{
  "systemChangeReqRef": "CR_RAN_LTE_20220512_090924.208",
  "recordTime": "2022-06-27T22:13:46.955781",
  "impactedData": [
    {
      "dataFieldNames": [
        "CELL","ALARM_INFO"
      ],
      "dataResourceNames": [
```

```
        "Alarm","NetworkMeasurement"
      ]
    }
  ],
  "impactedAnalytics": [
    {
      "dataResourceNames": [
        "Alarm","NetworkMeasurement"
      ],
      "dataComponents": [
        "DataExtraction","Clustering"
      ]
    }
  ]
}
```

Listing 2. An example of data quality impact change record

Strategies for specifying data quality contracts: To avoid a false signal that we have incomplete data for certain types of entities (e.g., the number of sites is different among different months) when utilizing a dataset, we must check if (i) the data of new entities has no effect on the analytics (e.g., when performing site traffic prediction) and (ii) the data of new entities has effect on the analytics (e.g., when performing a clustering for a network zone). We develop the two types of strategies that *data engineering operations* and *analytics operations* must consider: (i) data quality assurance for data processing activities and (ii) updating reference data used by analytics. The first type is involved in the development and improvement of `Data Processing Function`. The second one is a requirement for complex tasks of reexamining relevant data in `Analytics Reference Data` and, if needed, to retrain and update reference data. In the following, we discuss one example for the first strategy type.

We define and evaluate change metrics to characterize change phenomenons. A change metric has three factors:

- trend: indicates the direction of change, such as "increase" or "decrease"
- change rate: indicates the rate of changes to relevant analytics. Combining the change rate with the change time, we can establish conditions reflecting the change impact (e.g., high impact in case of a high change rate in a short change time).
- pattern: indicates the pattern of the change, such as sudden, incremental, and recurrent.

These factors are chosen due to their generality that we have seen in best practices to capture changes in data science/ML: change point detection (e.g., trend), operation reaction (e.g., change rate/impact), data and concept drift (e.g., change pattern). Change metrics are evaluated for a window of data. We use different functions for evaluating change metrics, such as:

- Trends are detected using trend estimation functions for data in a window. The type of data determines the type of the trend estimation function that a operator can use.
- We use a common way to determine change impact: $changerate = \frac{|y(t_e)-y(t_s)|}{y(t_s)}$, whereas t_e and t_s indicate the end time and start time (in day resolution for our use cases), respectively, of the change period for an entity and

$y(t_e)$ and $y(t_s)$ are the representative value of the entity at t_e and t_s respectively. Naturally, we can add and use new functions to determine change rates.

- Change patterns are detected using common pattern detection algorithms, especially for time series and drift [10, 31] and can be referred to by trend estimation functions.

Change metrics are applied only to selected entities under analytics (e.g., a zone or specific category of sites based on the same vendor with a similar configuration). Based on that, we specify constraints and actions in the strategy. Listing 3 shows an example of change impact metrics and constraints based on the TENSAI strategy specification documents in Figure 5.

```
{
"appliedTo": [
  {
    "dataResourceName": "NetworkMeasurement",
    "dataComponent": "metric_forecast_gm",
    "constraints": [
      {
        "dqMetricName": "changerate",
        "condition": {
          "operator": ">",
          "value": "2"
        }
      }
    ]
  }
],
"acceptedSolutions": [
  {
    "solutionName": "retrain_metric_forecast_gm",
    "description": "retrain"
  }
 ]
}
```

Listing 3. An example of metrics and constraints

Quality-aware data pipeline engineering: Due to different data storage strategies in V-RAN, it is not possible to correct all historical data in different datasets consisting of change entities. Therefore, along any data pipeline that handles data related to changing entities, software components must decide if the input data overlaps fully, partially, or none with the time and the list of changing entities to decide additional data correction. Figures 8 and 9 show high-level designs for a common data pipeline and a data quality impact change-aware data pipeline of software components and their configurations. Internally, inside appropriate components of the pipeline, the software implements features to deal with impact changes. Such features are designed in a generic way that accepts different configurations for

changes to automate additional corrections. From TENSAI, configurations can be generated (with/without input of data engineer/scientist) during *data engineering operations* or *analytics operations*. To support the engineering, TENSAI keeps the list of changes, including dates, and provides utilities for different situations. For example, if all data include changes, the correction might be done before merging as it saves time. Since identity change cannot be updated into data datalake/storage, it can be expensive to repeat the correction when the pipeline is executed often. In this case, we can decide to create new version of data and use new version of data as the input for the pipeline.

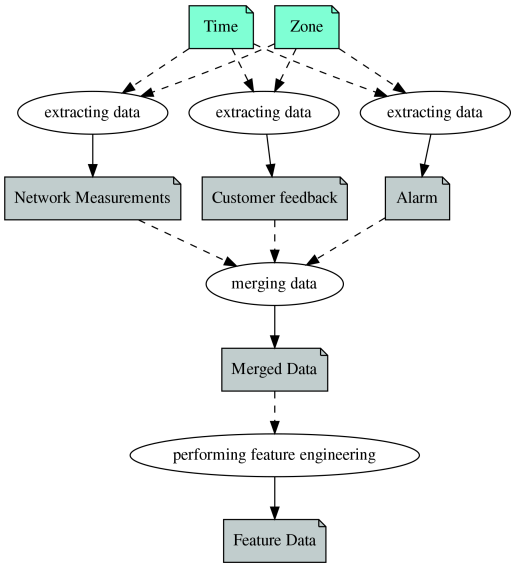

Fig. 8. Common data pipeline

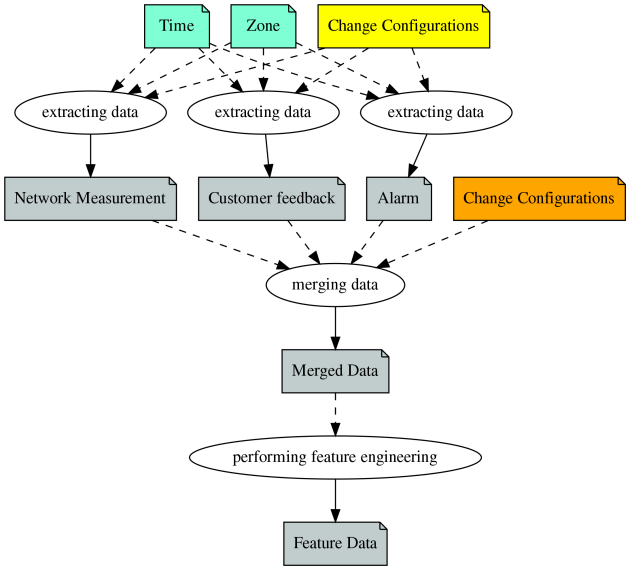

Fig. 9. Change-aware data pipeline with modified software components and configurations

One important aspect is to develop tools to detect change situations for metrics identified in the strategies that are suitable for V-RAN. For structured time series data, the fundamental software to be used is change point detection algorithms [5], pattern detection in timeseries [29, 44], and customized algorithms. For text data, currently we focus on rule-based and pattern detection due to our light, fast log preprocessing tool. In the future, we can investigate ML techniques, such as text/log classification and topic labeling [27] to decide if there are changes in the content. Existing software can automatically detect the differences but the software does not know which situations would mean entity changes in the specific context of the data. Therefore, we must customize code processing the output detected by existing software to determine the right metrics.

Another tricky situation is how to distinguish between change and missing data when the change is unknown by the people. In this case, we use automatic tools to detect the change but we require human operators to examine the result and determine if it is a problem of missing data or change. Figure 10 explains the combination of automatic detection of change in data with control configuration for data analytics. Software utilities can support automatic detection of change whereas humans will provide control configuration to decide stop or adjust the analytics.

4.1.3 Examples. Two common changes are changing the name and the location of sites/cells. In order to prepare network measurements for clustering, we need to check name identity change for network measurement data. This

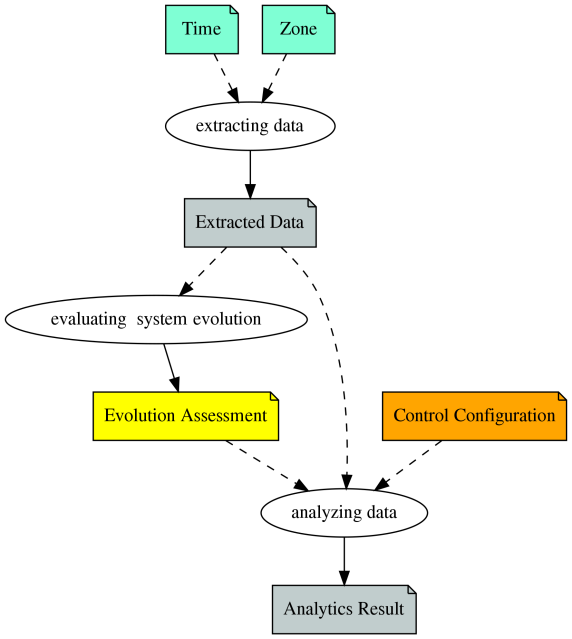

Fig. 10. Using additional check and control the analytics

can be done right after the data extraction. For example, we have changed more than 15000 cell/site entities. However, people in *data engineering operations* and *analytics operations* were not aware of the changes, until we detected some data problems. With TENSAI, the changes now can be available immediately for relevant operations. Given that `Data Extraction` has to support different requests, two possibilities for improving the data pipelines have been introduced: (i) updating all raw data before merging for different MLs or (ii) doing the update during the merging. A configuration change can be reflected as:

```
{
  "startTime": "2021-09-01 00:00:00",
  "endTime": "2021-12-31 23:59:59",
  "timeStampField": "DATE",
  "changeEntityMapping": [
    {
      "entityField": "CELL",
      "oldValueField": "Cellname",
      "newValueField": "Cellname_new"
    }
  ]
}
```

and change values can be stored in another dataset. Based on that, Listing 4 shows a simplified code abstracting the complex update of data based on configuration changes. For many changes, the update is just for well-defined data fields (e.g., based on column name of the data records). Therefore, we do not need to update the software pipeline but configurations, which can be generated from TENSAI and revised by the corresponding people.

```
#for pandas
def get_new_name(old_name):
    try:
        new_name =mapping_data[old_name][new_value_field]
        return new_name
    except:
        return old_name
data_df[entity_field_name]=data_df[entity_field_name].apply(lambda x:
    get_new_name(x))
#for spark
def get_new_name(old_name):
    if (old_name in cell_name_list.keys()):
        new_name=cell_name_list[old_name]
        return new_name
    return old_name
get_new_name_udf= udf(lambda x:get_new_name(x),StringType())
selected_data_df=data_df.withColumn(entity_field_name, get_new_name_udf(col(
    entity_field_name)))
```

Listing 4. Example of handling identity change for cell

In the second example, consider LC as a network zone representing a district. We detected its change (due to the evolution of the V-RAN infrastructure) using our detection tool that can be scheduled to run periodically. Figure 11 presents a result from change detection for LC. Information about the change can be reported to the operators, who can examine and confirm the situation. Listing 5 gives a concrete change confirmation message reported (see TENSAI message in Figure 4).

```
{
  "entity": {
    "entityType": "DISTRICT",
    "entityName": "LC"
  },
  "metrics": [
    {
      "metricName": "cell_evolution",
      "trend": "increase",
      "metricCategory": "evolution",
      "detectionMethod": {
```

```
        "methodName": "CUSUM",
        "detectionComponent": "ad_detect_evolution.py"
      },
      "changeLevel": "CELL",
      "startTime": "2021-10-25 00:00:00",
      "endTime": "****",
      "additionalData": [
        {
          "uri": "https://****.****.****",
          "desc": "figures in minio"
        }
      ],
      "confirmedStatus": "yes",
      "confirmedBy": "****",
      "confirmedTime": "****"
    }
  ]
}
```

Listing 5. Example of a change confirmation

Given the change confirmed, the information is propagated to `Data Scientist/Engineer` as well as other interested parties according to the communication and strategies defined in TENSAI. `Data Scientist/Engineer` use TENSAI to calculate the change metric for making decisions (need to retrain models or not). A TENSAI message about change metric is as follows:

```
{
  "messages": [
    {
      "trend": "increase",
      "changerate": "3.278",
      "pattern": "sudden",
      "startTime": "2021-09-01 00:00:00",
      "endTime": "2021-11-08 00:00:00",
      "dataResourceName": "NetworkMeasurement_LC"
    }
  ]
}
```

After evaluating the values in the change metric message, which recommends us to consider retraining models, `Data Scientist/Engineer` can continue to perform the data drift check to decide if existing ML models in `MachineLearning` must be retrained, thus being aware of possible drifts and proactive to prevent the ML model performance degradation.

For example, based on the data quality impact change message, consider the evolution of sites/cells during September-November, 2021. When we assume that the data from April-June, 2021 as reference data used for training ML models, we evaluated data drift for September-November, 2021 data. The result based on Evidently[14] reported "`Drift is detected for 70.83% of features (17 out of 24). Dataset Drift is detected`". The drift also showed clearly the drift w.r.t. the network downlink effect (new compared with the reference data) in Figure 12. Newly introduced cells have smaller download throughput due to their lower capacity configuration, compared to other cells. If the data drift due to the system evolution is unknown, ML models related to throughput analytics (e.g., classification, detection or prediction) can give misleading results (such as giving false alarms about network problems when actually there is no network issue). This forces `Data Scientist/Engineer` to retrain their ML models.

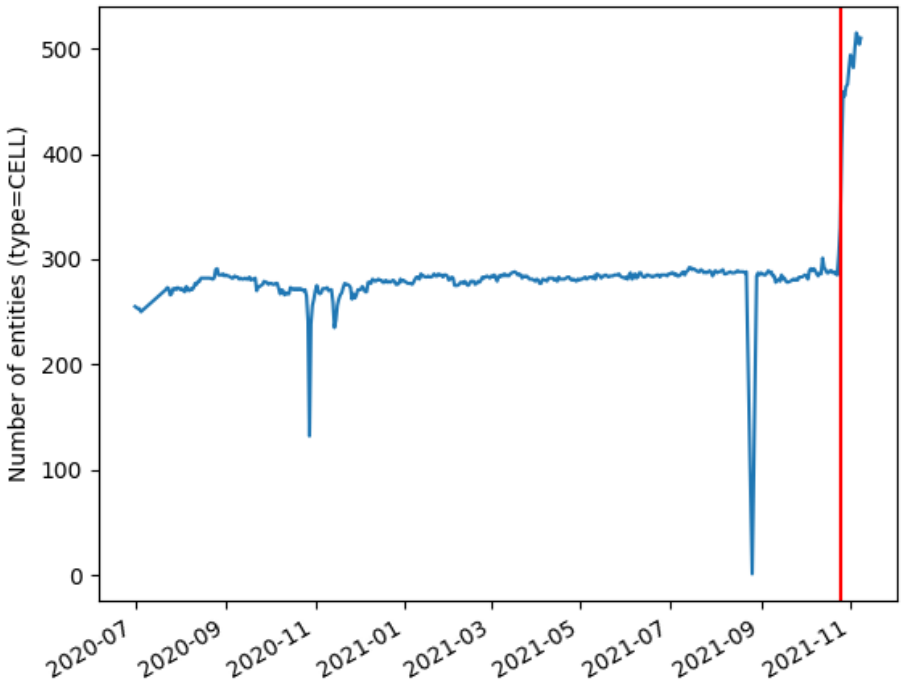

Fig. 11. Detection of the change of cells in a single district from a dataset of measurements with 6858308 records (including many districts). The vertical red line indicates the start time of the change (due to the evolution of the infrastructure).

4.2 Currency of data and its structure

4.2.1 Problem description. The currency of data and the data structure reflects how current our view on the data and its structure. For example, whether the data we have is current or old. Similarly, if we assume that the data structure is version $v1$, while in fact, the data structure is already in $v2$, we have a problem of currency in data structure. Both may

[14] https://github.com/evidentlyai/evidently

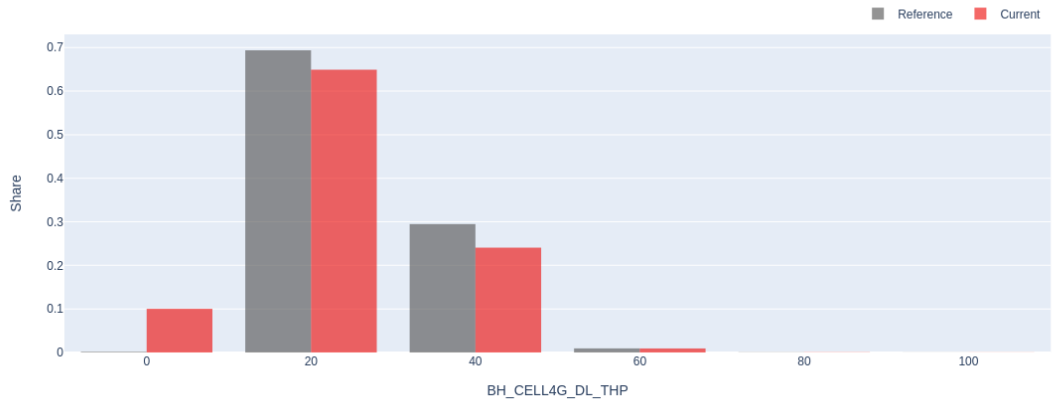

Fig. 12. Data drift of downlink network measurement, detected for LC, as a network zone, from a dataset of 6858308 records (including many districts)

lead to the problem of missing current data due to the unavailability of current data and the schema mismatch when processing data.

In data preprocessing for customer feedback, alarms, and incidents, we have parsed different types of logs. This is within the building block `Data Extraction` in Figure 1. They include both structured records and text logs in different languages (for example, Vietnamese and English in our datasets). The use of log parsing techniques [52] is just one part of the data preprocessing. However, a major of data quality problems is related to this part, such as unrecognized patterns lead to missing data. We have developed parsing modules and patterns to handle *current* raw formats defined by the system operator team or equipment vendors for operation purpose. However, the currency of data format/structure in the *analytics operations* is not the same as that in the *V-RAN operations*. It is often that software update and reconfiguration in *V-RAN operations* change the data patterns, such as timestamp and log structure, and create the currency problems w.r.t. data structures. Common data quality problems are log structure changes or the textual content of the data fields being updated (for example, adding or modifying the customer complaint root causes related to voice service when implementing IP Multimedia Subsystem system for Voice over LTE). It will lead to missing or incorrectly processed data in *analytics operations*.

4.2.2 *Dealing with data quality impact using TENSAI.*

Cause/effect identification and communication: One important aspect is to communicate if contents and structures of data (e.g., logs/alarms) have been changed. This often is due to hardware/software update and configuration changes in the system components where the data is originally measured/generated. The change could lead to a schema drift (the structure of the logs/alarms has been changed) and/or content drift. Therefore, the cause/effect of changes to the schema drift and content details in *data engineering operations* and data/concept drift in *analytics operations* must be defined. While in *V-RAN operations*, such changes may be known by `System Operator`, who manages the update, `System Operator` might not know the consequence of the changes on the schema drift or data drift, which are familiar concepts in *data engineering operations* and *analytics operations*. Enabling the communication of changes, TENSAI would fill the gap among these operations. Another type of information that needs to be communicated is the list of utilities and

tools for detecting potential changes in data schemas and content details. Such utilities and tools sample data to detect schemas and compare against existing schemas to detect the possibility of structure mismatch to present warnings for data engineering pipelines. *Strategies for specifying data quality contracts*: A requirement of detecting the currency problems must be defined. Strategies are defined for checking input data before performing other tasks. They can help to implement features to take constraints into steering further actions. Within an operation belonging to *analytics operations* or *V-RAN operations*, in order to solve an issue for an entity, the operation can identify required datasets as data resources and conditions that must be available for the operation. We determine *data currency* metrics for each dataset based on a requested window of available data $-w_d$, types of data $-dtype$, the issue – $issue$ – to be solved, and the operation – op. Based on state-of-the-art, we leverage existing data currency functions [15] and combine with ratio of the availability of data to calculate and manage currency: $datacurrency = (completeness, currency, issue, dtype, op, w_d)$. Let function $time()$ determine the timestamp and $latest()$ return the latest element, we have

- w_d includes data of $dtype$ returned from a request for available data from $t_s(w_d)$ to $t_e(w_d)$. t_s and t_e of w_d indicate the start and end time, respectively, for a time window length, $t_s \leq time(issue) \leq t_e$, identified by the operation for solving for $issue$ based on domain knowledge at $time(op)$. Given t_s and t_e, we can calculate the expected total length of records as $twl(dtype, issue)$ based on the measurement frequency of $dtype$. This applies only to Network Measurement. For other types of data, such as Alarm or Incident, $twl(dtype, issue) = len(w_d)$ as the data is event-based.
- $completeness = min(1, \frac{len(w_d)}{twl(dtype,issue)})$: measures the ratio between the amount of available data $-len(w_d)$ and the expected amount of data $-twl(dtype, issue)$ – required for the operation to be carried out.
- $currency = min(1, \frac{lagtime(dtype)}{time(op)-time(latest(w_d))})$: measures how current the available of data is for solving $issue$, based on the ratio between the optimal age of $dtype$ for the operation and the age of available data for $issue$. The optimal age is represented by $lagtime(dtype) > 0$ which is a known system parameter indicating the delay of the data due to system configuration impacting the readiness of the data. The age of the available of data is determined by the difference between $time(op)$ and the timestamp of the latest available data $time(latest(w_d))$[15]. $currency = 1$ means the highest currency of data. If there is no available data, $latest(w_d) == NONE$, the currency will be 0, the lowest value.

Given the currency and completeness of a datatype, the strategy specification will define possible actions such as which teams are suitable to take the next step to solve customer feedback or provide data currency information to data scientists or engineers performing related work. $datacurrency$ is context-specific for people/software carrying out the analytics of a given data and the operation. We have different operations with varying amounts of analytics for business, management, optimization, etc. purposes with different contexts. Each needs to deal with different data. Therefore, different `System Operator` and `Data Scientist` may have different values of data currency and they will have to act based on the values TENSAI calculates for them. Listing 6 shows an example of metrics and constraints based on $datacurrency$.

```
"appliedTo": [
{
"dataResourceName": "NetworkMeasurement",
"dataComponent": "PRButilization",
```

[15] In V-RAN we see that $time(op) > time(latest(w_d))$ as the query is faster than the update of data. In some rare situations, it is possible the query execution is delayed and the condition does not hold. This can be solved by adding a query delay.

```
"constraints": [
  {
    "dqMetricName": "completeness",
    "condition": {
      "operator": "<",
      "value": "0.95"
    }
  },
  {
    "dqMetricName": "currency",
    "condition": {
      "operator": "<",
      "value": "0.92"
    }
  }
]
}
],
"acceptedSolutions": [
{
"solutionName": "wait_for_constraints",
"description": "wait until the constraint is fulfilled."
}
],
```

Listing 6. An example of data currency and constraints

Quality-aware data pipeline engineering: Figure 13 gives a high-level view of addressing the data currency problems. At the basic level, a schema drift detection should be done within `Parsing data`. For example, in terms of Alarms, structure of records can be inferred. This can be done using sampling or sending the logs for checking schema drift. For example, alarm schemas can be represented in CSV. Based on CSV header detection and schema detection we can detect if schema drift has occurred. One important aspect is that the detection must be lightweighted and integrated to large-scale data processing technologies based on Apache Spark and Pandas. Another aspect is the incorporation of testing techniques, (such as in [23] and Validatar[16]) with the employment of `Analytics Reference Data`, as a lightweight monitoring/sampling error of preprocessing to report schema and data drift in `Error Assessment Report`. `Analytics Reference Data` includes key entities in the systems, such as a list of site/cell names, list of hardware components in system dependencies, and province/district. However, it is very challenging to test if the parsing is correct (e.g., extract the correct entity name) with millions of records and currently it is out of the scope of TENSAI. Based on `Error Assessment Report`, a decision will be made if we need to control `Parsing Data` or continue to `Analyzing Data`. Here `Error Assessment Report` is a means for communicating the quality to the next step in data

[16]https://github.com/yahoo/validatar

processing, such as realtime analytics or ML inferences. `Analyzing Data` also considers `Control Configuration`. Another problem is the data drift related to raw logs. While schema drift might be detected, changes of data detail of alarms are not easily detected and this requires new research.

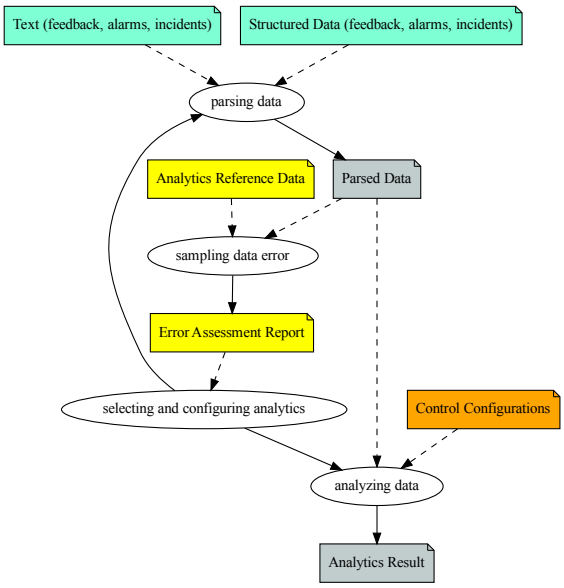

Fig. 13. Pipeline for addressing data currency problems

4.2.3 Examples. In our first example, considering the case of a customer service issue resolution in which TENSAI supports data currency for resolving the issue. A customer feedback records a possible site/cell/zone service issue that the customer has a problem. Resolving the issue may require the involvement of multiple types of operations across V-RAN. However, handling such information without understanding data currency and corresponding communication and strategies among these different operations could lead to huge effort and high cost. The call center (*business operations*) has no detailed network measurements or alarms to handle certain types of feedback. Thus, the feedback have to be passed to *V-RAN operations* due to data currency. Within *V-RAN operations*, there are different operations with different views on data and data currency. Consider the case with a site `Site69`[17] which was reported by customers at `t0=2022-06-17 22:15:13`. A quick check by the call center gave no answer to the problem with the customer's SIM card or data subscription. Thus, an issue due to the feedback was created and escalated to *V-RAN operations*:

```
{
    "feedback_id": "****",
    "received_time": "17/06/2022 22:15:13",
    "error_time": "17/06/2022",
```

[17]The site name has been changed to have an anonymous name to avoid revealing sensitive information.

```
    "feedback_category": "42_ACCESSING_DATA_ISSUE"
}
```

Three types of system operators in *V-RAN operations* are involved in this case: `Operator Type1` performs high-level checks of the customer's site serving, `Operator Type2` addresses site engineering and `Operator Type3` is network L1 optimizer/engineer[18], who can go to the site to get driving test or single-site verification test logfiles and access other low-level network data for in-depth analysis, such as PRB (Physical Resource Block) utilization. `Operator Type3` is only involved in handling emergencies or when other teams can't find the problem.

At time `t1=2022-06-17 22:33`, `Operator Type1` traced and found `Site69` as the serving cell and check historical KPIs in Network Measurement and alarm of `Site69` but no abnormalities are found. Therefore, `Operator Type1` used TENSAI to check *datacurrency* metrics with a requested data window ranging from `2022-06-17 00:00:00` to `2022-06-17 22:00:00` (based on `error_time` in the issue):

```
{
  "messages": [
    {
      "completeness": "1",
      "currency": "1",
      "dataResourceName": "Alarm",
      "datavisibility": [
        "Operator Type1",
        "Operator Type2",
        "Operator Type3"
      ]
    },
    {
      "completeness": "0.91",
      "currency": "1",
      "dataResourceName": "NetworkMeasurement-PRB",
      "datavisibility": [
        "Operator Type3"
      ]
    }
  ]
}
```

The above TENSAI's report on *datacurrency* indicates that (i) Alarm data is current for resolving the issue, while (ii) additional data (NetworkMeasurement-PRB) visible only for `Operator Type3` does not meet the condition (*completeness* = 0.91 vs *condition* = 0.95). Since `Operator Type1` cannot find any problem with `Site69` and the severity of feedback is still low, the issue was propagated to `Operator Type2` with additional information about `Site69` as the serving cell for checking neighboring sites to localize the fault area.

[18]Network L1 (Layer1) is about the physical layer

At time t2=2022-06-18 07:43:02, using Uber H3[19] resolution 8, Operator Type2 finds three sites in the same area {Site69,Site36,Site59}. Carrying out analytics with network measurements and alarms, Operator Type2 found nothing. Operator Type2 did the same thing with resolution 7 with 5 sites: {Site10, Site04,Site69,Site70,Site42}. Although Site04 has some alarms, these did not signal any suspicion. At this time, different datasets are accessed by *V-RAN operations* and the *datacurrency* for the datasets related to op=feedback(42_ACCESS_DATA_ISSUE) is calculated by TENSAI is:

```
{
  "messages": [
    {
      "completeness": "1",
      "currency": "1",
      "dataResourceName": "Alarm",
      "datavisibility": [
        "Operator Type1",
        "Operator Type2",
        "Operator Type3"
      ]
    },
    {
      "completeness": "1",
      "currency": "1",
      "dataResourceName": "NetworkMeasurement-PRB",
      "datavisibility": [
        "Operator Type3"
      ]
    }
  ]
}
```

To resolve the issue, given *datacurrency* met the condition but Operator Type2 does not have access to NetworkMeasurement - PRB, Operator Type2 decided to escalate to another operation within *V-RAN operations* carried out by Operator Type3. At time t3=2022-06-18 14:48:57, Operator Type3 received feedback information and used the same way with zone analytics to access on a new type of data PRB and the data of alarms and PRB are more current (due to a delay of data update (2-3 hours) and the gap between t2,t3). First, the value of RPB of Site69 signaled that Site69 was overloaded. Given the strategy of data currency, Operator Type3 must look for other close sites and alarms to find the root cause.

```
DATE                    ALARM INFO
17/06/2022 20:47:36     Resource status indication, cell disabled
17/06/2022 20:47:36     Resource status indication, cell disabled
```

[19] https://h3geo.org/

```
17/06/2022 20:47:36    Resource status indication, cell disabled
```

Such alarms, happened before `t0`, triggered the examine of other information and the root cause was founded by `Operator Type3`: `Site04` had connection problems and was not able to serve customers, leading to traffic handovers to neighbour sites and `Site69` became overloading to cause problems for customers.

Fig. 14. Reconstructed important network measurements over the time for `Site69` and `Site04`. The data was based on peak hours to demonstrate the behavior of sites. The operators got the data analytics based on the data at the analytics time only

Our second example is related to the currency of data schema w.r.t. the data detail (content of log). In *V-RAN operations*, `System Operator` can swap RAN equipment for many reasons (such as old equipment has been depreciated and needs to be replaced or launching equipment modernization projects), which can lead to a change in alarm detail and/or format and affect data preprocessing for analytics. The following example describes the alarm format of two vendors, `Vendor N` and `Vendor E`, in the same region at two different times (before and after the swap):

```
#alarm format vendor N
"7652|BASE STATION NOTIFICATION","Failure in internal BTS connection or
    connection to 3rd party tool||6261 unitName=FSMF path=/SMOD_R-1 serial_no=
    L1170804346 dstIPAddr=****:**** additionalFaultID=6261 SMOD-1||||"
#alarm format vendor E
"Sync Frequency PDV Problem","High PDV detected on IP packets towards PTP
    Grandmaster#-ProbableCause(OSS)=m3100Unavailable",
```

These alarm data will enter into the same data streams and will be ingested and processed by a component (preprocessing) and the result will be fed into other analytics. Due to the structure of the data (CSV, text), the change might not be detected until the analytics time. The orginal data preprocessing pipleline supports the first schema, thus, it can

handle the record well (such as, able to obtain `"unitName": "FSMF","dstIPAddr":"****", "FaultID": 6261`). Due to the problem of data format/detail currency, the preprocessing will not be able to parse the record, thus yield no result. This failure would also mean that data/concept drift, which are important for ML, might not be detected. Why rigorous schema drift detection has been extensive researched [13], detecting changes in data detail is not easy and under developed, given complex data within V-RAN. For example in terms of log details, we can combine the schema detection (such as `bigquery-schema-generator`[20] for CSV and JSON data) with well-known schema drift detection techniques. For detecting the change of data content detail due to *V-RAN operations* of reconfiguring/updating software, we measure/sample the error when using existing grok patterns with contents. If the error rate is higher than a threshold defined by `Data Quality Manager` based on the domain knowledge, the error report will be forwarded to V-RAN operations team to confirm if there was any change in the system that made the error and trigger the review process of logs by data engineering using TENSAI communication messages. This way can be replaced by advanced ML such as text classification[21], which would introduce more overhead for the sampling process. This technique is also used for anomaly detection of log details due to security breach or rare events. Besides, the error report is also used to decide if the related analytics could be stopped in districts with a high error rate or should be switched to other algorithms automatically or done by a data scientist team.

4.3 Missing data

4.3.1 Problem description. Missing data is a common problem and it affects several analytics in V-RAN. Our focus is to detect missing data and to control subsequent actions to react on the selection, configuration and execution of analytics based on missing data (but not handling missing data within specific analytics as it is another subject of research and engineering). Thus, we deal with missing data for inputs of specific analytics in the view of the operators, who interpret missing data according to their operation context. Concretely, the severity of data missing can be evaluated to support data quality-aware control in `Data Extraction` or `ChangeDataCapture` processes, which provide data for specific analytics. This also means that data quality control is focused on concrete metrics to avoid to work on generic metrics that may have no impact on the analytics. This differs from dealing with missing data in data storage or data ingestion (where the data can be used for different analytics).

Missing data can happen at two levels: record level (missing the whole records) and missing fields (within records). Missing data can be due to system errors and/or input errors (e.g., operators do not enter the data). However, the impact of missing data would be evaluated for specific analytics within a context of an operation. In Figure 1, each indicate in Network Measurement is determined from many counters associated with many different elements and factors. In V-RAN, we have thousands of counters for 2-5G in different measurement period (e.g., event-based, minutes, hours, and daily). When collecting these counters from `OSS:Monitoring` to the `NM:DB` through `ChangeDataCapture`, missing data issues happen sometimes due to many reasons (synchronization loss, some components not working well or error, change source, etc.). The key issue is that missing data cannot be detected easily by examining the analytics results. It may also be too late to detect at the analytics as due to complex underlying systems (like `OSS:Monitoring`), the original data may not be available after a period of time due to the system capability (e.g., 7 days). Therefore, handling missing data must be implemented in the pipeline if required.

4.3.2 Dealing with data quality impact using TENSAI.

Cause/effect identification and communication: First, we define clearly data quality metrics related to missing data. Such

[20] https://github.com/bxparks/bigquery-schema-generator
[21] https://developers.google.com/machine-learning/guides/text-classification

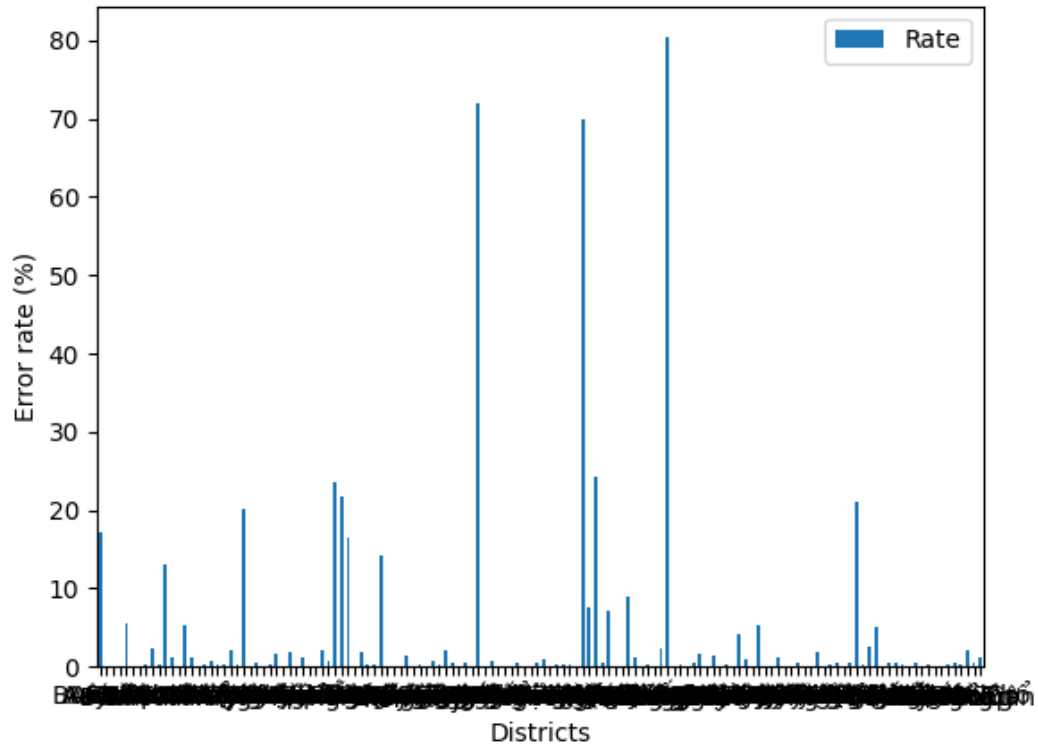

Fig. 15. Error rate of processing alarm logs in districts with a dataset for 9 months (1140268 log records). Data schemas for logs in few districts have been changed a lot. A consequence is that the target ML analytics may have a problem.

metrics are associated with types of expected inputs for specific analytics, indicating the role of missing data for such analytics. Due to the domain requirement, TENSAI helps communicate clearly the importance of data quality at the records and analytics subjects levels (such as cell/site/zone and its measurements) but not at the data field level within records (as this level is not relevant in V-RAN operations). From the communication perspective, analytics algorithms are also documented with missing data handling capabilities (based on various literature [9]). Thus, as a consequence of missing data, if we could not handle missing data, we still have a good understanding of which analytics (and its internal algorithms) can handle missing data well. Therefore, the communication has to document well the algorithms associated with analytics with expected data quality.

Strategies for specifying data quality contracts: Concretely, in our strategies we define (i) data completeness ($datacompleteness$) metrics at record levels for site/cell measurements (missing measurement records) and for zone (missing cells/sites) and (ii) data dispersion among sites/cells for zone-based analytics (using standard deviation), for example:

- record-level missing data as $datacompleteness = 1 - \frac{count(missingrecords)}{total}$

- dispersion by value as $datadispersion = std(x(en_i))$ vs by time as $datadispersion = std(x(en_i, t_j))$ where std is the standard deviation function, x is the number of records for entity en_i either for all times or at a specific time t_j.

In the context of V-RAN, if the timeliness, validity, and accuracy of the data are not met, we consider them causing missing data (as either the data is not available or cannot be used and can be removed like `NaN` values). We define constraints in strategy specifications, in which data completeness and data dispersion metrics must be met for appropriate analytics and types of analytics with suitable internal algorithms are accepted solutions given the values of metrics. These constraints can be updated. However, they have to be setup based on expertise and domain-knowledge for types of data and analytics. This will be defined by `Data Quality Manager` and `Data Scientist/Engineer` with the input from the user of analytics.

Quality-aware data pipeline engineering: Figure 16 shows an example that the output of extracting data (e.g., in `ChangeDataCapture`) and the result is used to perform data profiling to decide which algorithms should be run (or should not), if missing data is detected or extracting data must be redone to update data before running analytics. TENSAI communicates data quality metrics, together with the type of data (network measurement) as input for the reaction to be carried out automatically or manually (via strategies). In order to make the decision, the error rate must be estimated by training a prediction model using data quality metrics and error rates and suitable invocations of analytics/ML are implemented based on constraints of the error rates. The constraints can also specify (additional) algorithms that impute missing data [1] to provide estimated true measurements for missing ones. Which methods should be used are not in the scope of TENSAI, which enables the communication and integration of data quality with possible consequent actions by leveraging other MLs.

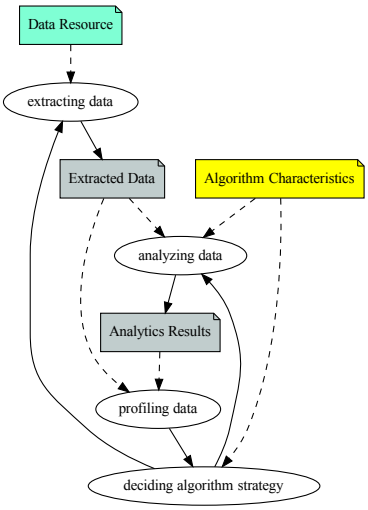

Fig. 16. Handling missing data in analytics

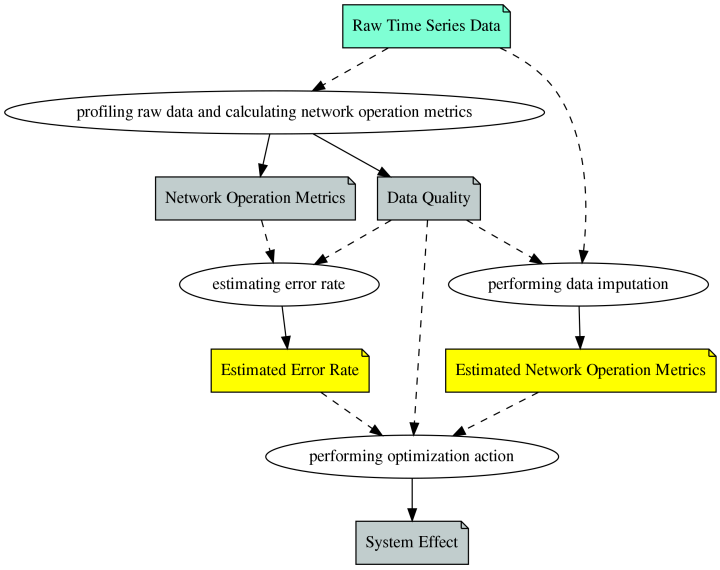

Fig. 17. Data quality-aware system control/optimization given missing data

4.3.3 Examples. We illustrate two examples to show diverse impacts of missing data to different analytics. Consider *CSFB_SR* (Circuit Switched Fallback Success Rate) for a zone, site or cell as a key metric for evaluating voice services when customers have bad experiences (and for other optimization). This metric is monitored continuously via common analytics like realtime streaming analytics with Apache Flink/Apache Spark[22] and t-digest [8] for different window times (hourly and daily). The metric will be retrieved by `System Operator` to consider if, for example, voice services cannot be switched from 4G to 3G and the system does not support voice over LTE (VoLTE), given complaints from the customer. This metric is calculated as $CSFB\_SR = 100\% * \frac{CSFB\_Success}{CSFB\_Request}$. The *CSFB_Success* and *CSFB_Request* values are aggregated based on appropriate time windows (hourly or daily) from the records of the 15 minutes measurement frequency. When one or more records are missing, *CSFB_SR* value is not calculated correctly. Detecting this problem cannot be done by examining the analytics result, e.g., shown in Table 3 where with two data quality metrics – *completeness* and *datadispersion* – high missing data rates and data dispersion could lead to higher *CSFB_SR*. This causes a big problem for reactive actions of `System Operator` in *V-RAN operations* at realtime. Furthermore, since aggregated *CSFB_SR* will be stored for long-term prediction using ML algorithms, data problems will be propagated to data lake/storage. Therefore, in our strategies `ChangeDataCapture` and analytics components implements the data quality monitoring and check against with expected constraints. For example `System Operator` in *V-RAN operations* expected to have max 0.01% error rate. The data quality will be used to trigger quality assurance and enforce exactly-once data policies from `OSS:Monitoring`. Furthermore, for daily *CSFB_SR*, data quality will be stored to provide additional information for `MachineLearning`. Thus through measuring and providing data quality metrics, we use them for communicating potential problems to different stakeholders, of which involvements spread over different pipelines at different times.

Data quality	CSFB_SR	Error rate
'completeness': 0.1, 'datadispersion': 0.46	99.877	0.017
'completeness': 0.2, 'datadispersion': 0.52	99.684	0.177
'completeness': 0.3, 'datadispersion': 0.7	**99.904**	**0.044**
'completeness': 0.4, 'datadispersion': 0.77	99.812	0.048
'completeness': 0.5, 'datadispersion': 0.8	99.877	0.017
'completeness': 0.6, 'datadispersion': 0.9	99.886	0.026
'completeness': 0.7, 'datadispersion': 0.78	99.852	0.008
'completeness': 0.8, 'datadispersion': 0.93	99.844	0.017
'completeness': 0.9, 'datadispersion': 0.65	99.868	0.008
'completeness': 1.0, 'datadispersion': 0.0	99.860	0.000

Table 3. Example of *CSFB_SR* for a single zone with 10 sites for single day. Missing data is introduced by using sampling techniques of Pandas.

The second example is to consider data as input for important ML-based analytics, such as predicting data traffic growth for network planning and evolution. Consider a *V-RAN operation* to explore traffic growth in a specific zone. The zone is defined by a distance from a specific location. In our test, we select a location and use Uber H3 resolution = 7 (average hexagon area $\tilde{5}.16$ km^2 and average hexagon edge length 1.22 km) and predict different V-RAN traffic types, including 2-4G. We emulated missing data missing and observed the effect of missing data on the prediction. Figure 18 shows one example of the effect of missing data with 4G traffic. Clearly, predicted values are smaller and the

[22]https://flink.apache.org/ and https://spark.apache.org/

peak/busy hours are different. Thus, being unaware of missing data could lead to a wrong decision. In this situation, by integrating TENSAI features into the data pipeline, TENSAI will provide data quality metrics for deciding if an ML algorithm should be invoked. TENSAI strategies can also provide insightful information to help select or trigger suitable ML algorithms with right parameters (such as for tackling missing data).

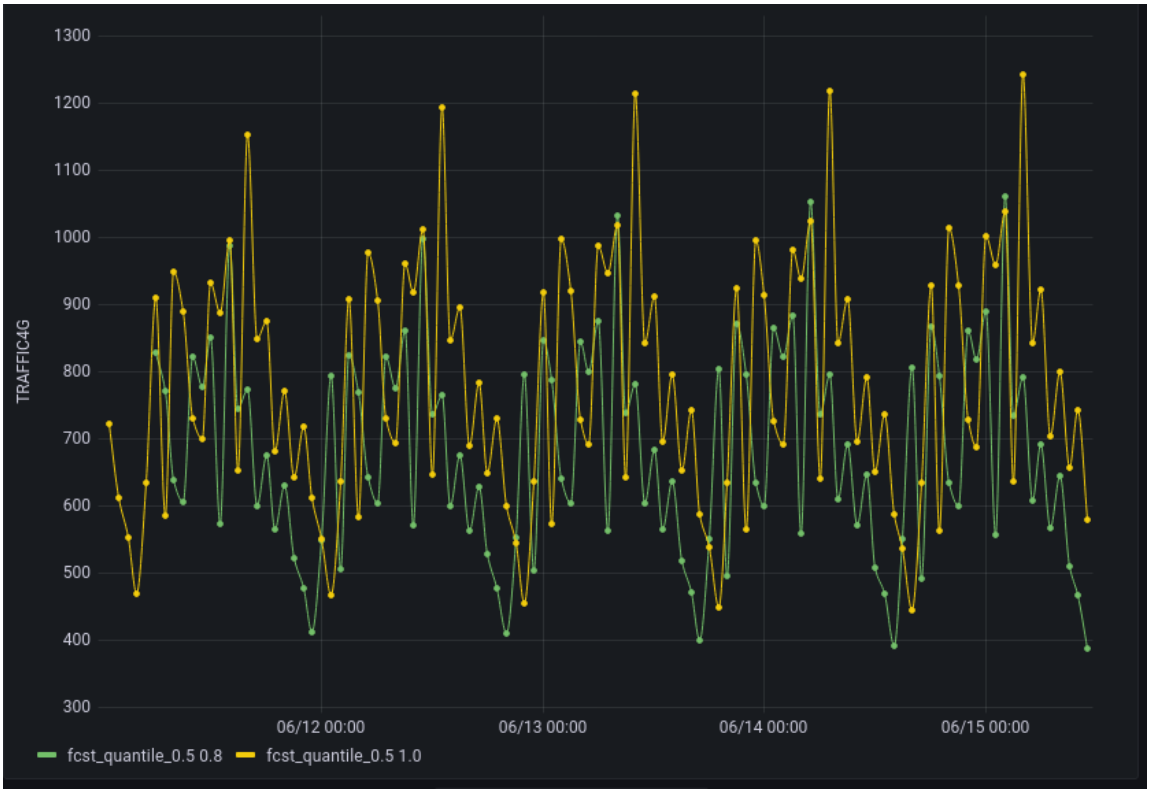

Fig. 18. Forecast of traffic 4G using Kats Global Model [39]: quantile 0.5 with $datacompleteness == 1$ (line `fcst_quantitle_0.5 1.0`) and $datacompleteness == 0.8$ (line `fcst_quantitle_0.5 0.8`)

Note that some prediction algorithms cannot work if timeseries data is missing. In this case, detecting data problems by TENSAI would prevent the execution of the prediction and support automatically switching of prediction algorithms. Thus, the strategy will help to use and verify the result. Furthermore, we note that it is possible to carry out data imputation with suitable libraries or with suitable parameter configurations for an ML algorithm. However, these libraries must also be tested and put into the strategies. For example, using Luminaire[23] in a simple way, we observed some low accuracy of interpolated values for missing data:

```
                     raw  interpolated
2022-05-09 02:00:00 5.59577  5.595770
2022-05-09 03:00:00 NaN      4.153613
2022-05-09 04:00:00 1.90175  1.901750
```

[23] https://zillow.github.io/luminaire/

```
...                     ...         ...
2022-06-15 19:00:00 33.38404  33.38404
2022-06-15 20:00:00 NaN      36.971412
2022-06-15 21:00:00 42.16067  42.16067

                     raw
2022-05-09 03:00:00 1.93416
2022-06-15 22:00:00 32.05771
```

This example demonstrates that strategies should also consider the variety of algorithms abilities in selecting and invoking suitable algorithms. TENSAI helps to manage such complex relationships between data quality constraints and algorithm abilities.

5 RELATED WORK

In mobile networks, there are many big data analytics and ML case studies for various problems, such as [14, 51]. Our work in this paper is not focused on such analytics per see, but on the question of how to facilitate the utilization and correctness of complex big data analytics by filling the gaps in communication, strategy specifications and engineering of data quality awareness for such analytics. There is no lack of papers about data quality in general and data quality in telecommunications and in data analytics [25, 42]. Many of these works just discuss the methodology at the level of conceptual high-level frameworks, raising key metrics and possible approaches but without concrete techniques and practical solutions for large-scale mobile networks. This is partially due to the fact that solving data quality problems one must work (i) with concrete data in domain contexts and (ii) with pipelines and interpretation of data quality based on specific operations. In our work, we apply existing methodologies to define/evaluate some concrete metrics for demonstrating our solutions. From the domain understanding, the work in [4] provides some high-level recommendations for designing big data quality in different situations that we take into account, e.g., we consider to link the original sources of data problems to all possible subsequent actions on data using communication, strategies and engineering. In the view of data science, papers like [40] emphasize the lifecycle management. While our work does not work on lifecycle for specific data science projects or datasets, we address the lack in establishing a uniform view on several aspects of data. However, we look at cross systems with many phases among different types of operations beyond the boundaries of a lifecycle.

Recently, DataOps approaches have incorporated automation for data lifecycle engineering and management [30], dealing also with data quality within a pipeline [38]. TENSAI can be integrated with DataOps and vice versa. However, our TENSAI goal is different as it works for cross systems with different types of data, introducing communication/strategy and engineering techniques to achieve the goal, where automation features could be incorporated but are not in our focus. Furthermore, DataOps solutions usefully work well for a single type of data within a single system, but not among different types of data spreading different systems where automation might not be fully established in a cross-system manner. In software engineering, organization management and information systems, there are many papers discussing about change management [6, 21]. They provide foundation concepts for changes and communications of changes. Our work differs by focusing on change and strategy associated with data quality for big data analytics.

At the technical level, TENSAI relies on different enablers for evaluating data quality problems. Different techniques show how to evaluate quality of data using different tools and integration models [43, 46]. Various techniques are for

anomaly detection [16, 34, 41] that can be used to detect changes in data. Many open sources of anomaly detection tools, such as Twitter AD[48], Alibi [49], Kats [19], EGADS [20] can be integrated into data analytics frameworks. Similarly, one can easily follow many ML models for time series data [24, 33]. DataOps-4G is a platform supporting data quality discovery [50]. Our work in this paper is to leverage existing tools and develop customized utilities that can be integrated into the quality process. In [35] the paper discusses quality data evaluation for ML, following the data science lifecycle. Our TENSAI is not bound to a data science cycle, which can be contained within certain activities in our work. In this paper we did not work on the data quality for ML algorithms. However, we share a common view on the importance of quality evaluation. We address the strategies in configuring and executing data pipelines and their possible embedded ML methods. But we do not work at the level of ML algorithms.

In terms of exchange data quality impacts and problems, works like [12] present a data quality model used to inform the user. But such a work does not answer the question of how to detect and adapt software based on data quality. Most research and industrial observability systems are not focused on data quality [22, 32]. As a framework, TENSAI supports data observability approach [28], a subject which has been increasingly discussed. Both TENSAI and data observability services, stimulated by software and service observability but targeting to data, must support key data quality metrics that have been well studied. But TENSAI is more than data observability techniques, which are strongly related to data quality evaluation in TENSAI. TENSAI supports communication and strategies associated with observed data quality metrics to support the different types of operations, including data analytics tasks. TENSAI does not focus on solving data problems found in data, which are the main goals of data observability. Naturally, if a suitable data observability service exists for V-RAN, TENSAI can utilize such a service.

6 CONCLUSIONS AND FUTURE WORK

Efficient ways to integrating data observability, data quality impacts, and suitable actions in large-scale analytics across multiple systems and operations, like in V-RAN, require us to capture quality impact cause/effect, communicate data quality problems, develop strategies and engineer data quality-aware pipelines. We have presented TENSAI framework as a set of practical ways to deal with data quality impacts in very large analytics infrastructures. We show that, for such complex systems, we must incorporate different techniques from data engineering, software development and team collaboration. The rich, available set of algorithms and techniques to detect data quality and to analyze data under different quality awareness levels must be selected and used in a coherent view of data quality constraints based on operations and business contexts. The TENSAI framework has presented different solutions for the real-world data analytics based on concrete techniques to convey quality problems and potential impacts from their original sources along data engineering and analytics pipelines for different operations. Most of datasets used for our proof-of-concept are for 3G and 4G networks, the most active usage in our V-RAN with 2-5G technologies. However, there should be no difference when applying TENSAI to operations requiring 2G or 5G datasets.

Although TENSAI is developed to deal with strategies and actions centered around data quality problems and consequences, we see that the data quality problem is just one excellent candidate to demonstrate hard problems of doing data science across different types of operations in a large-scale system. Therefore, we believe that TENSAI can be extended for other problems, such as the quality of analytics results from ML models and performance of ML models. Furthermore, this problem is investigated with our mobile networks data, but it can be applied to other domains, such as electricity networks and manufacturing.

Our future work is focused on engineering and integration of TENSAI into complex business and operation workflows in V-RAN. Further characteristics of data detection techniques, data quality-aware ML algorithms, and communication

means will be incorporated to provide richer data quality assurance strategies and actions. Automation based on data quality metrics and strategies will be the next step.

DECLARATIONS

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Availability of data and materials

Not applicable

Competing interests

The authors declare that they have no competing interests.

Funding

Not applicable

Author's Contributions

HLT identified topics and led the work, designed the work, developed the prototype, identified data, carried out experiments, and edited the manuscript. NNNT identified topics, identified data and examples, developed the prototype, carried out experiments and edited the manuscript.

Acknowledgements

We are grateful to the support from Le Xuan Duc and Tran Huu Dat from Mobifone for the data collection and extraction work.

REFERENCES

[1] Deepak Adhikari, Wei Jiang, Jinyu Zhan, Zhiyuan He, Danda B. Rawat, Uwe Aickelin, and Hadi A. Khorshidi. 2022. A Comprehensive Survey on Imputation of Missing Data in Internet of Things. *ACM Comput. Surv.* (apr 2022). https://doi.org/10.1145/3533381 Just Accepted.

[2] Sylvio Barbon Junior, Gabriel Marques Tavares, Victor G. Turrisi da Costa, Paolo Ceravolo, and Ernesto Damiani. 2018. A Framework for Human-in-the-Loop Monitoring of Concept-Drift Detection in Event Log Stream. In *Companion Proceedings of the The Web Conference 2018* (Lyon, France) *(WWW '18)*. International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 319–326. https://doi.org/10.1145/3184558.3186343

[3] Carlo Batini, Cinzia Cappiello, Chiara Francalanci, and Andrea Maurino. 2009. Methodologies for Data Quality Assessment and Improvement. *ACM Comput. Surv.* 41, 3, Article 16 (jul 2009), 52 pages. https://doi.org/10.1145/1541880.1541883

[4] David Becker, Trish Dunn King, and Bill McMullen. 2015. Big data, big data quality problem. In *2015 IEEE International Conference on Big Data (Big Data)*. 2644–2653. https://doi.org/10.1109/BigData.2015.7364064

[5] Gerrit J. J. van den Burg and Christopher K. I. Williams. 2020. An Evaluation of Change Point Detection Algorithms. https://doi.org/10.48550/ARXIV.2003.06222

[6] Rune Todnem By. 2005. Organisational change management: A critical review. *Journal of Change Management* 5, 4 (2005), 369–380. https://doi.org/10.1080/14697010500359250 arXiv:https://doi.org/10.1080/14697010500359250

[7] Corinna Cortes, L. D. Jackel, and Wan-Ping Chiang. 1994. Limits on Learning Machine Accuracy Imposed by Data Quality. In *Advances in Neural Information Processing Systems*, G. Tesauro, D. Touretzky, and T. Leen (Eds.), Vol. 7. MIT Press.

[8] Ted Dunning. 2021. The t-digest: Efficient estimates of distributions. *Software Impacts* 7 (2021), 100049. https://doi.org/10.1016/j.simpa.2020.100049

[9] Tlamelo Emmanuel, Thabiso M. Maupong, Dimane Mpoeleng, Thabo Semong, Banyatsang Mphago, and Oteng Tabona. 2021. A survey on missing data in machine learning. *J. Big Data* 8, 1 (2021), 140. https://doi.org/10.1186/s40537-021-00516-9

[10] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2019. Deep learning for time series classification: a review. *Data Min. Knowl. Discov.* 33, 4 (2019), 917–963. https://doi.org/10.1007/s10618-019-00619-1

[11] Amirata Ghorbani and James Zou. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, 2242–2251.

[12] Eliza Gyulgyulyan, Julien Aligon, Franck Ravat, and Hrachya Astsatryan. 2019. Data Quality Alerting Model for Big Data Analytics. In *New Trends in Databases and Information Systems*, Tatjana Welzer, Johann Eder, Vili Podgorelec, Robert Wrembel, Mirjana Ivanović, Johann Gamper, Mikoaj Morzy, Theodoros Tzouramanis, Jérôme Darmont, and Aida Kamišalić Latifić (Eds.). Springer International Publishing, Cham, 489–500.

[13] Hossein Hamooni, Biplob Debnath, Jianwu Xu, Hui Zhang, Guofei Jiang, and Abdullah Mueen. 2016. LogMine: Fast Pattern Recognition for Log Analytics. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management* (Indianapolis, Indiana, USA) *(CIKM '16)*. Association for Computing Machinery, New York, NY, USA, 1573–1582. https://doi.org/10.1145/2983323.2983358

[14] Ying He, Fei Richard Yu, Nan Zhao, Hongxi Yin, Haipeng Yao, and Robert C. Qiu. 2016. Big Data Analytics in Mobile Cellular Networks. *IEEE Access* 4 (2016), 1985–1996. https://doi.org/10.1109/ACCESS.2016.2540520

[15] Bernd Heinrich and Mathias Klier. 2011. Assessing data currency — a probabilistic approach. *Journal of Information Science* 37, 1 (2011), 86–100. https://doi.org/10.1177/0165551510392653 arXiv:https://doi.org/10.1177/0165551510392653

[16] Bilal Hussain, Qinghe Du, and Pinyi Ren. 2018. Deep Learning-Based Big Data-Assisted Anomaly Detection in Cellular Networks. In *2018 IEEE Global Communications Conference (GLOBECOM)*. 1–6. https://doi.org/10.1109/GLOCOM.2018.8647366

[17] Abhinav Jain, Hima Patel, Lokesh Nagalapatti, Nitin Gupta, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Ruhi Sharma Mittal, and Vitobha Munigala. 2020. Overview and Importance of Data Quality for Machine Learning Tasks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining* (Virtual Event, CA, USA) *(KDD '20)*. Association for Computing Machinery, New York, NY, USA, 3561–3562. https://doi.org/10.1145/3394486.3406477

[18] Georgios Kathareios, Andreea Anghel, Akos Mate, Rolf Clauberg, and Mitch Gusat. 2017. Catch It If You Can: Real-Time Network Anomaly Detection with Low False Alarm Rates. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 924–929. https://doi.org/10.1109/ICMLA.2017.00-36

[19] Kats. 2022. *Kats*. https://github.com/facebookresearch/Kats

[20] Nikolay Laptev, Saeed Amizadeh, and Ian Flint. 2015. Generic and Scalable Framework for Automated Time-series Anomaly Detection. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1939–1947.

[21] Steffen Lehnert. 2011. A Taxonomy for Software Change Impact Analysis. In *Proceedings of the 12th International Workshop on Principles of Software Evolution and the 7th Annual ERCIM Workshop on Software Evolution* (Szeged, Hungary) *(IWPSE-EVOL '11)*. Association for Computing Machinery, New York, NY, USA, 41–50. https://doi.org/10.1145/2024445.2024454

[22] Bowen Li, Xin Peng, Qilin Xiang, Hanzhang Wang, Tao Xie, Jun Sun, and Xuanzhe Liu. 2022. Enjoy your observability: an industrial survey of microservice tracing and analysis. *Empir. Softw. Eng.* 27, 1 (2022), 25. https://doi.org/10.1007/s10664-021-10063-9

[23] Nan Li, Anthony Escalona, Yun Guo, and Jeff Offutt. 2015. A Scalable Big Data Test Framework. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*. 1–2. https://doi.org/10.1109/ICST.2015.7102619

[24] Markus Löning, Anthony J. Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J. Király. 2019. sktime: A Unified Interface for Machine Learning with Time Series. *CoRR* abs/1909.07872 (2019). arXiv:1909.07872 http://arxiv.org/abs/1909.07872

[25] Jorge Merino, Ismael Caballero, Bibiano Rivas, Manuel Serrano, and Mario Piattini. 2016. A Data Quality in Use model for Big Data. *Future Generation Computer Systems* 63 (2016), 123–130. https://doi.org/10.1016/j.future.2015.11.024 Modeling and Management for Big Data Analytics and Visualization.

[26] Microsoft. 2022. Data contracts. https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/scenarios/cloud-scale-analytics/architectures/data-contracts

[27] Marcin Michał Mirończuk and Jarosław Protasiewicz. 2018. A recent overview of the state-of-the-art elements of text classification. *Expert Systems with Applications* 106 (2018), 36–54. https://doi.org/10.1016/j.eswa.2018.03.058

[28] Barr Moses. 2022. The Rise of Data Observability: Architecting the Future of Data Trust. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining* (Virtual Event, AZ, USA) *(WSDM '22)*. Association for Computing Machinery, New York, NY, USA, 1657. https://doi.org/10.1145/3488560.3510007

[29] Abdullah Mueen. 2014. Time series motif discovery: dimensions and applications. *WIREs Data Mining and Knowledge Discovery* 4, 2 (2014), 152–159. https://doi.org/10.1002/widm.1119 arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1119

[30] Aiswarya Raj Munappy, David Issa Mattos, Jan Bosch, Helena Holmström Olsson, and Anas Dakkak. 2020. From Ad-Hoc Data Analytics to DataOps. In *Proceedings of the International Conference on Software and System Processes* (Seoul, Republic of Korea) *(ICSSP '20)*. Association for Computing Machinery, New York, NY, USA, 165–174. https://doi.org/10.1145/3379177.3388909

[31] Hai-Long Nguyen, Yew-Kwong Woon, and Wee Keong Ng. 2015. A survey on data stream clustering and classification. *Knowl. Inf. Syst.* 45, 3 (2015), 535–569. https://doi.org/10.1007/s10115-014-0808-1

[32] Sina Niedermaier, Falko Koetter, Andreas Freymann, and Stefan Wagner. 2019. On Observability and Monitoring of Distributed Systems – An Industry Interview Study. In *Service-Oriented Computing*, Sami Yangui, Ismael Bouassida Rodriguez, Khalil Drira, and Zahir Tari (Eds.). Springer International Publishing, Cham, 36–52.

[33] A. Nielsen. 2019. *Practical Time Series Analysis: Prediction with Statistics and Machine Learning*. O'Reilly Media, Incorporated.

[34] Md Salik Parwez, Danda B. Rawat, and Moses Garuba. 2017. Big Data Analytics for User-Activity Analysis and User-Anomaly Detection in Mobile Wireless Network. *IEEE Transactions on Industrial Informatics* 13, 4 (2017), 2058–2065. https://doi.org/10.1109/TII.2017.2650206

[35] Hima Patel, Nitin Gupta, Naveen Panwar, Ruhi Sharma Mittal, Sameep Mehta, Shanmukha Guttula, Shashank Mujumdar, Shazia Afzal, Srikanta Bedathur, and Vitobha Munigala. 2022. Automatic Assessment of Quality of Your Data for AI. In *5th Joint International Conference on Data Science amp; Management of Data (9th ACM IKDD CODS and 27th COMAD)* (Bangalore, India) *(CODS-COMAD 2022)*. Association for Computing Machinery, New York, NY, USA, 354–357. https://doi.org/10.1145/3493700.3493774

[36] Leo L. Pipino, Yang W. Lee, and Richard Y. Wang. 2002. Data Quality Assessment. *Commun. ACM* 45, 4 (apr 2002), 211–218. https://doi.org/10.1145/505248.506010

[37] Neoklis Polyzotis, Martin Zinkevich, Sudip Roy, Eric Breck, and Steven Whang. 2019. Data Validation for Machine Learning. In *Proceedings of Machine Learning and Systems*, A. Talwalkar, V. Smith, and M. Zaharia (Eds.), Vol. 1. 334–347.

[38] Sagar Sen, Erik Johannes Husom, Arda Goknil, Simeon Tverdal, Phu Nguyen, and Iker Mancisidor. 2022. Taming Data Quality in AI-Enabled Industrial Internet of Things. *IEEE Software* (2022), 0–0. https://doi.org/10.1109/MS.2022.3193975

[39] Slawek Smyl. 2020. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting* 36, 1 (2020), 75–85. https://doi.org/10.1016/j.ijforecast.2019.03.017 M4 Competition.

[40] Victoria Stodden. 2020. The Data Science Life Cycle: A Disciplined Approach to Advancing Data Science as a Science. *Commun. ACM* 63, 7 (jun 2020), 58–66. https://doi.org/10.1145/3360646

[41] Kashif Sultan, Hazrat Ali, and Zhongshan Zhang. 2018. Call Detail Records Driven Anomaly Detection and Traffic Prediction in Mobile Cellular Networks. *IEEE Access* 6 (2018), 41728–41737. https://doi.org/10.1109/ACCESS.2018.2859756

[42] Ikbal Taleb, Mohamed Adel Serhani, Chafik Bouhaddioui, and Rachida Dssouli. 2021. Big data quality framework: a holistic approach to continuous quality management. *J. Big Data* 8, 1 (2021), 76. https://doi.org/10.1186/s40537-021-00468-0

[43] Hui Yie Teh, Andreas W. Kempa-Liehr, and Kevin I-Kai Wang. 2020. Sensor data quality: a systematic review. *J. Big Data* 7, 1 (2020), 11. https://doi.org/10.1186/s40537-020-0285-1

[44] Sahar Torkamani and Volker Lohweg. 2017. Survey on time series motif discovery. *WIREs Data Mining and Knowledge Discovery* 7, 2 (2017), e1199. https://doi.org/10.1002/widm.1199 arXiv:https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1199 e1199 DMKD-00255.R2.

[45] Hong-Linh Truong, Marco Comerio, Flavio De Paoli, G.R. Gangadharan, and Schahram Dustdar. 2012. Data contracts for cloud-based data marketplaces. *International Journal of Computational Science and Engineering* 7, 4 (2012), 280–295. https://doi.org/10.1504/IJCSE.2012.049749 arXiv:https://www.inderscienceonline.com/doi/pdf/10.1504/IJCSE.2012.049749 PMID: 49749.

[46] Hong-Linh Truong and Schahram Dustdar. 2010. On Evaluating and Publishing Data Concerns for Data as a Service. In *2010 IEEE Asia-Pacific Services Computing Conference*. 363–370. https://doi.org/10.1109/APSCC.2010.54

[47] Hong Linh Truong and Tri-Minh Nguyen. 2021. QoA4ML - A Framework for Supporting Contracts in Machine Learning Services. In *2021 IEEE International Conference on Web Services, ICWS 2021, Chicago, IL, USA, September 5-10, 2021*, Carl K. Chang, Ernesto Daminai, Jing Fan, Parisa Ghodous, Michael Maximilien, Zhongjie Wang, Robert Ward, and Jia Zhang (Eds.). IEEE, 465–475. https://doi.org/10.1109/ICWS53863.2021.00066

[48] Twitter. 2022. *AnomalyDetection R package*. https://github.com/twitter/AnomalyDetection

[49] Arnaud Van Looveren, Janis Klaise, Giovanni Vacanti, Oliver Cobb, Ashley Scillitoe, Robert Samoilescu, and Alex Athorne. 2019. *Alibi Detect: Algorithms for outlier, adversarial and drift detection*. https://github.com/SeldonIO/alibi-detect

[50] Shaochen Yu, Tianwa Chen, Lei Han, Gianluca Demartini, and Shazia Sadiq. 2022. DataOps-4G: On Supporting Generalists in Data Quality Discovery. *IEEE Transactions on Knowledge and Data Engineering* (2022), 1–1. https://doi.org/10.1109/TKDE.2022.3151605

[51] Kan Zheng, Zhe Yang, Kuan Zhang, Periklis Chatzimisios, Kan Yang, and Wei Xiang. 2016. Big data-driven optimization for mobile networks toward 5G. *IEEE Network* 30, 1 (2016), 44–51. https://doi.org/10.1109/MNET.2016.7389830

[52] Jieming Zhu, Shilin He, Jinyang Liu, Pinjia He, Qi Xie, Zibin Zheng, and Michael R. Lyu. 2019. Tools and Benchmarks for Automated Log Parsing. In *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*. 121–130. https://doi.org/10.1109/ICSE-SEIP.2019.00021