Journal of Data Intelligence, Vol. 3, No. 2 (2022) 218–231

THE IMPACT OF DATA COMPLETENESS AND CORRECTNESS ON EXPLAINABLE MACHINE LEARNING MODELS

SHELERNAZ AZIMI
Free University of Bozen-Bolzano
39100 Bolzano, Italy
fname.sname@unibz.it

CLAUS PAHL
Free University of Bozen-Bolzano
39100 Bolzano, Italy
fname.sname@unibz.it

Many systems in the Edge Cloud, the Internet-of-Things or Cyber-Physical Systems are built for processing data, which is delivered from sensors and devices, transported, processed and consumed locally by actuators. This, given the regularly high volume of data, permits Artificial Intelligence (AI) strategies like Machine Learning (ML) to be used to generate the application and management functions needed. The quality of both source data and machine learning model is here unavoidably of high significance, yet has not been explored sufficiently as an explicit connection of the ML model quality that are created through ML procedures to the quality of data that the model functions consume in their construction. Here, we investigated the link between input data quality for ML function construction and the quality of these functions in data-driven software systems towards explainable model construction through an experimental approach with IoT data using decision trees. We have 3 objectives in this research: 1. Search for indicators that influence data quality such as correctness and completeness and model construction factors on accuracy, precision and recall. 2. Estimate the impact of variations in model construction and data quality. 3. Identify change patterns that can be attributed to specific input changes. This ultimately aims to support *explainable AI*, i.e., the better understanding of how ML models work and what impacts on their quality.

Keywords: Explainable AI, AI Engineering, Data Quality, IoT Systems, Machine Learning, Data Correctness, Data Completeness, Decision Trees.

1. Introduction

There are different types of errors or faults which may occur in data sets, such as missing values or rows, invalid values or formats, or duplicated values or rows. Low quality data will result in low quality machine learning models if the model is used to learn from the data. Before using often faulty real world data and trying to find a remedial solution for observed machine learning model, we need to better understand the effects of low input data quality on the created models.

Our ultimate goal is to automate quality control of machine learning models, but to reach that the understanding the impact of a sensor producing faulty data or no data on a model trained on this data is a general requirement. The wider objective is explainable model construction. Black-box explainable AI aims at a better understanding of how ML model output depends on the model input [20]. Of particular importance is here a root cause

analysis for model deficiencies. Our aim here is, based on observed model quality problems, to identify a root cause at input data level. The concrete practical benefit of this in an IoT setting for example is, that certain ML quality patterns might already point to specific problems with the data, such as outages for faulty sensors.

Therefore, we investigated different experimental scenarios with artificial and real faulty input data sets. We specifically considered 1) input data completeness and 2) input data correctness, since these are of direct relevance to IoT settings. With the experiments, we created situations with different faulty data sets and compare the results to find a connection between the type of faulty data and the ML quality assessment factors (accuracy, precision, recall). We focus here on numeric data that would for example be collected in technical or economic applications, neglecting text and image data here. This paper extends the earlier [4] by technical details and experimental results. Furthermore, we have positioned this in the context of *Explainable AI* and *AI Engineering*.

The novelty lies in the integrated investigation the quality of information that is derived from data through a machine learning approach. We proposed a quality frameworks in [2], [1], but report on an in-depth experimental study here.

The remainder of the paper is structured as follows. In Section 2, we provide some background on Explainable AI. In Section 3, we review related work. Our methods, experiments and comparisons are presented in Section 4. Sections 5 and 6 present the experimental analysis and summarise the evaluation results, before concluding in Section 7.

2. Explainable AI

Explainable AI is the context of this work. Explainability is the extent to which the internal mechanics of a machine or deep learning system can be explained in human terms. Interpretability is about being able to identity the mechanics without necessarily knowing why. Explainability is being able to explain what is happening. Our ultimative objective is to automate a root cause analysis that aims to 'explain' the reasons for quality deficiencies or defects [13] in the ML model. This explains ML quality in terms of data quality [17].

Applied to our Internet-of-Things (IoT) setting, this means that for instance accuracy problems with traffic or weather prediction models or often cause by either unsuitable ML model construction or by data quality problems of data that is processed by the ML models. Here, we are specifically interested in understanding the impact of IoT data quality concerns. This in concrete terms meaning to understand if sensor failures cause incorrect readings or if network outages cause the data to be incomplete.

Overall, the aim is to move towards an explainability or interpretability of ML model failures/deficiencies as an a-posteriori measure for detection and correction [31]. Pre-construction data validation is an advisable step prior to model construction. In contrast to works in this context, we aim to identify missing values/default replacements as the root cause of prediction deficiencies (such as accuracy) as a remedial action. Some problems will still go undetected in an a pre-construction approach. Our approach (an a-posteriori analysis) can be adjusted to the presence of a-priori validation of data. Our approach also allows a black-box mode, if the construction itself is not visible/observable.

AI Engineering [6] is working towards a systematic construction of for instance ML models in order to achieve and maintain quality. Our root cause analysis can also been seen as an

endeavour to continuously improve quality.

3. Related Work

Machine learning (ML) techniques have generated huge impacts in a wide range of applications such as computer vision, speech processing, health or IoT.

Input data quality is important. The issue of missing data is unavoidable in data collection [7], [22], [15], [32]. Various imputation approaches, i.e., substituting missing values, have been proposed to address the issue of missing values in data mining and machine learning applications. [22] addresses missing data imputation. The authors propose a method called DIFC integrating decision tress and fuzzy clustering into an iterative learning approach in order to improve the accuracy of missing data imputation. They demonstrated DIFC robustness against different types of missing data.

Currently, missing data impacts negatively on the performance of machine learning models. Regarding concrete ML techniques, handling missing data in decision trees is a well studied problem [11]. [33] also proposed a method for dealing with missing data in decision trees. In [15], authors tackle this problem by taking a probabilistic approach. They used tractable density estimators to compute the "expected prediction" of their models. Missing data or uncertain data in general have always been a central issue in machine learning and specially classifiers. [32] focused on the accuracy of decision trees with uncertain data. The authors discovered that the accuracy of a decision tree classifier can be improved if the complete information of a data item is utilized. They extended classical decision tree algorithms to handle data tuples with uncertain data. Paper [26] describes a solution pattern that analyzed IoT sensor data and failure from multiple assets for data-driven failure analysis. The paper used univariate and multivariate change point detection models for performing analysis and adapted precision, recall and accuracy definition to incorporate the temporal window constraint. In [28], a toolkit for structured data quality learning is presented. They defined 4 core data quality constructs and their interaction to cover the majority of data quality analysis tasks.

Focusing on decision trees and missing data, we investigate the link between source data and machine learning model as a so far unexplored AI explainability concern.

4. Method

Before presenting the results of the experiments in the following section, we introduce here our methods including the description of objectives, data and implementation.

4.1. *Objectives*

In many applications, ML models are reconstructed continuously based on changing input data. We use experiments to determine the extent to which different input changes regarding data quality impact on model construction quality. In more concrete terms, the question is if changes in the data quality or the model construction have a similar impact on output quality. We consider here the following ML quality attributes. *Precision*, also known as Positive Predictive Value (PPV), answers the question of how many selected items are relevant. *Recall*, or Sensitivity, answers the question of how many relevant items were selected. *Accuracy* is

the percentage of correct predictions for the test data.

For input data quality, we selected two attributes that are IoT-relevant [3]: **completeness** is the degree to which the number of data points required to reach a defined accuracy threshold has been provided and **correctness** is the degree to which data correctly reflects an object or an event described, i.e., how close a label is to the real world.

In the context of these definitions, a sample question is if minor changes in the completeness of data (as a data quality problem) or the tree depth of decision trees (as a model construction concern) have a similar impact on model accuracy. Experiments shall help to determine the scale of the impact of a given size on input variations. We use experiments to determine if certain input change patterns correlate to observable output change patterns [12]. In concrete terms, this is if minor or major changes in input and input quality result in identifiable change patterns across different output qualities (e.g., accuracy, precision, recall). The question is if observed change patterns in the ML model output can be attributed to the root cause of that change at input data level.

4.2. *Implementation and Data Sets*

Our models here are decision trees – using scikit-learn[a] to both data sets for predictions. Using traffic data, see Fig. 1, we predicted the traffic volume and using weather data we predicted rain fall. The first data set was traffic data that has been taken from an application, which consisted of daily averages of traffic and number of vehicles in 72 stations around our province in a month. The total number of rows in this data set is thus 72.

Station Code	Actual detec	Average daily light		heavy	motorcycles	cars and smal	cars and small var	vans and mini	light trucks	heavy trucks	trucks with trailer	articulated trucks	buses	Target
00000001	478	8064	7103	960	411	6013	127	552	247	257	122	291	43	2
00000002	1483	13542	12163	1379	375	10420	113	1255	469	344	140	416	9	3
00000003	1325	21885	20537	1349	776	18483	139	1140	528	358	123	309	30	3
00000004	1236	19028	17689	1339	495	14872	157	2165	467	337	132	364	39	3
00000005	1100	8608	7850	707	226	6961	104	560	272	166	67	112	90	2
00000006	1435	14315	13572	737	322	12730	17	504	243	270	86	56	83	3
00000007	1407	15473	14714	758	764	12883	94	973	332	164	37	81	144	3
00000008	1391	5652	5241	411	335	4352	50	504	193	115	18	49	36	1
00000009	924	7410	7144	266	390	6195	44	516	133	59	9	14	51	2
00000010	1373	5383	5129	254	425	4007	42	654	176	35	10	27	6	1

Table 1. Traffic Data Set - Selection.

The second data set, see Fig. 2, was weather data consisting of the minimum and maximum temperature, rainfall, wind speed, humidity, pressure, cloud and rain today as features, and the target is the possibility of rain fall the next day for 49 stations.

Location	MinTemp	MaxTemp	Rainfall	WindSpeec	WindSpeec	Humidity9	Humidity3	Pressure9a	Pressure3p	Cloud9am	Cloud3pm	Temp9am	Temp3pm	RainToday	Target
1	9	14,3	7	11	20	93	80	1010,8	1010,3	8	8	11,1	11,3	1	1
1	3,6	14,5	4,2	9	19	90	55	1023,3	1022,9	0	0	8,2	13,8	1	0
1	1,1	14,1	0,2	4	15	100	49	1018,8	1017,2	7	6	3,9	13,1	0	0
1	3,9	10,9	0	6	0	88	82	1020,5	1018,8	7	8	6,4	8,8	0	1
2	8,3	14,5	17,4	17	19	96	90	1021,2	1020,7	0	0	10,9	13,3	1	1
2	9,3	19	0	13	19	81	64	1028,1	1027,3	0	0	13,5	17,8	0	1
2	12,2	16,2	23,6	13	11	99	83	1032,6	1030,9	0	0	13,8	15,3	1	0
2	0,8	18,6	0,2	0	20	99	40	1018,8	1015,5	0	0	6,9	17,8	0	0
3	11,9	20,1	14,2	9	17	87	72	1020,9	1017,1	8	2	16,1	19,8	1	1

Table 2. Weather Data Set - Selection.

The data from both data sets consisting of only numerical values has been processed and labeled manually.

4.3. *Experiments*

[a]https://scikit-learn.org/ - Machine learning library for Python.

Table 3. Incompleteness and Incorrectness Experiments Summary.

	Completeness	**Correctness**
Rows	In the traffic data, precision and recall behaved slightly different from accuracy but we do not see the same behavior for weather data. However, there is no significant difference.	For -1000 the values fell from lower initial values than in traffic data. For -5000, accuracy, precision and recall fell but the gradient was steeper than for -1000. For -10000, all three factors fell from a lower initial value but the final values are not lower than before. Therefore, the graph gradient is slighter when in fact the higher invalid value has effected the factors correctly.
Features	The stable area in the accuracy graph in the missing row does not occur in for missing features, where we see a soft fall. For the precision and recall, the sudden rise does not occur here. All factors have a steady gradient not as steep as for missing rows.	Accuracy is gradually falling, but precision and recall are acting differently. There is no connection to previous cases as those were from missing rows and invalid features here. Comparing the results we can say that this results are more understandable to the lower invalid value results because like there, accuracy is showing a steep and steady fall where on the other hand precision and recall are acting differently in a more unpredictable way.

The experimental strategy was to find the effect on accuracy, precision and recall while inducing error into the data set. We start each experiment with an initial baseline for these quality attributes. In order to check the impact of incomplete and incorrect input data on accuracy we created two different situations for each data set. For *incompleteness*, we checked the impact of *Missing Features* and *Missing Rows* on accuracy, precision and recall. For *incorrectness*, we checked the impact of *Invalid Features* and *Invalid Rows* for different invalid values on accuracy, precision and recall.

The experiments on input data completeness and incorrectness have been summarised in Table 1. For each data set in each table, we performed the experiments in two different formats, missing or invalid rows and missing or invalid features. The values were selected to reflect small, medium and large scale faulty situations. The values are in that sense meaningful in relation to the size of the data set in rows or features. For the missing or invalid rows in traffic data, we started with 2 rows and increased the number of missing rows gradually to 5, 15 and 24. For the missing or invalid features, we started with 3 features then 7 then 10 and lastly 13 features. For the missing or invalid rows in weather data we started with 6 rows and increased the number of missing rows gradually to 20, 36 and 49. For the missing or invalid features we started with 2 features then 4 then 8 and lastly 13 missing features. We observed the accuracy, precision and recall in these situations with 20% test size and tree depth of 3.

A summary of the results is presented in Fig. 3. The details of these observations follow then in Figs. 4 to 6.

For the weather data set we tried another set of invalid values as well to test the accuracy of the machine learning tool in identifying invalid values. As we mentioned before, in the first set we tried negative values as clearly invalid, but in the second set we tried extreme positive values as potentially possible, though highly improbable rainfall values. In general, we wanted to reflect different categories of sensors values: (i) correct sensor readings within small sensor reading variation, (ii) extreme but in principle possible values, likely linked to sensor faults, and (iii) clearly incorrect reading, definitely linked to sensor faults. We are dealing with

Table 4. Incorrectness Experiments Summary (Negative Rows).

	Traffic	**Weather**
Negative Invalid Rows	**Overall Results:** Accuracy, precision and recall all decreased with increasing the invalid rows but after 15 rows they all started to increase again. A possible explanation is over-fitting. **Observations:** According to the graphs what we can state is that from 5 invalid rows until 24 all three factors follow the same pattern but for smaller numbers of invalid rows the results vary. **Interpretation:** The results for more significant errors were as expected.	**Overall Results:** We started with -1000 as the incorrect value. With increasing the invalid rows, accuracy, precision and recall were gradually reduced. **Observations:** The accuracy fell from 65 to 35, precision fell from 64 to 41 and recall fell from 64 to 45 which all of them are lower values compared to the traffic data. **Interpretation:** There is a gradual fall of all three factors. For -5000, there is a steeper and more steady graph with increasing the invalid value.
	Comparison: For -1000 it needs to be pointed out that the values fell from lower initial values than in the traffic data. For -5000, accuracy, precision and recall fell but the gradient was steeper than the -1000 one. For -10000, all three factors fell from a lower initial value but the final values are not lower than the previous situations. Therefore, the graph looks softer when in fact the higher invalid value has effected the factors correctly.	

sensor data and chose invalid values that are out of the range of regular sensor readings. We generally chose 3 different incorrect settings in order to avoid unexpected behaviour from a single invalid value – typically choosing a clearly incorrect value such as -1000 and increasing this to the next order of magnitude. What we are also looking for is to find out which type of invalid values (positive or negative) can be identified better by the machine learning tool, thus allowing a better judgement of the possible root causes. The same experiments were repeated also on positive values. Compared to the negative results no significant pattern changes were identified except that the output values were less in positive values.

After observing the effect of different levels of faulty situations on accuracy, precision and recall, the next step was to try to find a concrete change pattern on each outcome factor's variation in different scenarios in order to connect those patterns to a specific scenario. To do so, we also tested the effect of different tree depths and different test sizes on normal and various faulty data sets and compare the results with each other in order to find a specific change pattern. We present the results in Table 2.

5. Observation, Analysis and Validation

The outcome of the experiments demonstrated similarity between the data sets and thus a validity of the observations as they have been confirmed in two settings.

5.1. *Experiments and Observations*

In total, we conducted more than 50 experiments that varied settings in 4 dimensions (tree depth, test size, missing/invalid features, missing/invalid rows). The respective settings and

Table 5. Incorrectness Experiments Summary (Negative Features).

	Traffic	**Weather**
Negative Invalid Features	**Overall Results:** The results showed a steady fall for accuracy but precision and recall acted differently. **Observations:** Accuracy, precision and recall fell significantly and this fall got steeper with higher values until -1000. From -1000 the results started to change, accuracy, precision and recall first fell but afterwards they started to increase. This situation got steeper with higher values. **Interpretation:** For precision and recall, the graph rose after an initial fall. The results changed when we changed the invalid value to a higher value.	**Overall Results:** Starting from -1000, with increasing the invalid features, the accuracy, precision and recall gradient falls gradually. **Observations:** In -5000 the accuracy, precision and recall fell more significantly than for -1000. The gradient is steeper. And finally for -10000, the graphs are steady and they fall without any change. **Interpretation:** There is not much difference between invalid rows and invalid features.
	Comparison: We are facing a familiar behavior where the accuracy is gradually falling but precision and recall are acting differently but there is no connection since the previous one was from missing rows and here, we have invalid features. Comparing this outcome to the lower invalid value results, the diagram for all three factors became a steep and somehow steady fall. Comparing the results we can say that this results are more understandable to the lower invalid value results because like there, accuracy is showing a steep and steady fall where on the other hand precision and recall are acting differently in a more unpredictable way.	

their observations are illustrated in Figs. 1 to 4.

As a summary of the findings, we can state that:

1. *Incorrectness more significant than Incompleteness.* The incorrectness has a bigger effect on the accuracy than the incompleteness. The most probable reason for it is that in incompleteness the machine learning tool may ignore the missing rows or features and not engage them in the predictions and calculations, but regarding incorrectness the tool is forced to use all the values either correct or incorrect therefore it cannot control or minimize the damage to the accuracy.

2. *Rows more significant than Features.* Missing or invalid rows have a stronger impact on the accuracy than missing or invalid features. Here again, the causes may be different factors, but the most probable one may be the fact that dealing with a complete missing or invalid row is more difficult than dealing with some missing or invalid features. Remedying the reduction of accuracy is more difficult with missing or invalid rows than missing or invalid features, see Figure 1.

3. *Data set differences.* In the analysis of the experiments, we noted that the results of the weather data was easier to process than the traffic data. In the traffic data set, the volume of data might have been rather low.

Table 6. Incorrectness Experiments Summary (Positives).

	Results
Positive Invalid Rows	**Overall Results (1000):** Compared to the negative values the accuracy did not change much and decreased only a few points but as for precision and recall the drop is more considerable. **Observations:** Accuracy fell from 57 to 30, precision fell from 65 to 31 and finally recall fell from 60 to 31. **Interpretation:** Overall compared to the traffic data, all three factors are lower. **Overall Results (5000):** With increasing the invalid rows the accuracy, precision and recall fell. **Observations:** Accuracy fell from 53 to 28, precision fell from 54 to 27 and recall fell from 54 to 28 which all of them either compared to the negative values or traffic data are lower. **Interpretation:** Compared to 1000's results, a significant fall in the results is visible but we can see that the gradient's steepness is similar to 1000 because the fall is based on the same ratio. In general, the results are similar to the negative values and lower than the traffic data. **Overall Results (10000):** Everything was as expected. Accuracy, precision and recall fell from lower values to even lower values and compared to previous positive value, negative values and traffic data, the results were lower.
Positive Invalid Features	**Overall Results (1000):** All three factors fell as expected. **Observations:** the accuracy fell from 68 to 40, precision fell from 67 to 39 and recall fell from 67 to 43. **Interpretation:** Compared to the invalid rows the results are higher but compared to the same experiment with negative values and traffic data set, the outcomes are lower. **Overall Results (5000):** The results are similar to the invalid rows. From 6 rows to 20 rows, we observed a sudden fall followed by a gradual fall afterwards but in the invalid features, the fall was gradual from the beginning. **Observations:** Accuracy fell from 55 to 30, precision fell from 53 to 30 and recall fell from 52 to 30. **Interpretation:** Compared to the negative values, the steepness of the graph is not as expected. Compared to the 1000, we noted a fall in the values but not in graph steepness, compared to the negative values, the results were lower and in overall compared to the traffic data, the results were lower. **Overall Results (10000):** We noted an initial rise in precision and recall. The accuracy for invalid features is very similar to the invalid rows either in values or graph but overall the accuracy in invalid rows is lower than invalid features. The same can be said about the precision and recall even though the graphs look different, but overall the values are lower for invalid rows than invalid features. Compared to the negative values, both results look less steep, but they are lower. The same can be said in comparison to the traffic data.

4. *Overfitting.* As a general observation, with very high results in the outcome, we tend have a machine learning tool problem like over-fitting, but when we have low results in outcomes, it means that the problem lies more likely in the data or sensors.

5. *Incorrect and Improbable Data.* Regarding positive and negative values, i.e., highly improbable vs. certainly incorrect data, we observed for weather data that the results for positive invalid values were lower than negative invalid values. This situation needs to be tested on other data sets to determine a reason. However, for weather data and with some negative values as inputs, a plausible explanation is that it is difficult to identify a real negative error, but for positive values, since the values were very high, it was easier for the algorithm to identify them.

In conclusion, the observations are validated in both data sets and are practically applica-

Table 7. Comparison Summary (TS: Test Size, TD: Tree Depth).

Rows-TD	In traffic data, the accuracy fell with increasing the missing rows. Depths 3, 4 and sometimes 5 were the best and anything below or over were unstable. This was shown better in the weather data set. The accuracy increased until the depth of 3, 4 and sometimes 5 and then started to fall which is expected. In traffic data, the accuracy first increased with tree depth but from the depth 3 to 5 was stable and after that fluctuated irregularly. In weather data, a similar result is visible. The best accuracy was at depths 3 to 5 as well but afterwards the accuracy started to fall. The fall was more significant with higher incorrectness.
Features-TD	In both data sets, accuracy started to rise until depth 4 and afterwards to fall. However, in traffic data it started to grow again after depth 8. A probable reason is over-fitting. In traffic data the accuracy rose from depth 1 to 3-4, then varies and then after reaching the depth 8 it rose again. In weather data, the accuracy rose from depth 1 to 4 and then it fell significantly. In traffic data, the first rise is expected because it's normal for accuracy to rise until the best depth but the second rise is due to a machine learning tool error or over-fitting.
Row-TS	In traffic data, accuracy falls with more missing rows but improves with bigger test sizes. The best test sizes were 20% and 30%. For weather data, accuracy improved until 20% and 30% before falling again. In traffic data, accuracy gradually increased until 30% but varied afterwards. In Weather Data, the results were more clear. The accuracy first rose until the best test size and then started to fall gradually. The best test sizes were 20% and 30%.
Features-TS	The best test size for both data sets were 20% and 30%. In traffic data, accuracy started to grow after 40% but according to the other experiments and weather results, probable reasons are ML errors or over-fitting. The results were similar to the previous experiments. Overall, the effect of invalid features on accuracy was less than the effect of invalid rows.

ble in machine learning quality analysis. They can be used in root cause analyses to identify possible faults in a IoT architecture such as sensor or connectivity problems. This provides a post-hoc explanation to black-box explainable model construction and recommendation of remedies [8, 9].

However, a clear identification of the reason behind the observation is not always possible. The problem here is the *white-box explainability* of machine learning models. As deep learning and other highly accurate black-box models develop, the social demand or legal requirements for interpretability and explainability of machine learning models are becoming more significant [27]. Nowadays, the two terms are beginning to have different meanings, with interpretability describing the fact that the model is understandable by its nature (e.g. decision trees) and explainability corresponding to the capacity of a black-box model to be explained using external resources (e.g., visualizations). However, white-box explainability is beyond the scope of the paper here.

6. Validation

We used two data sets to investigate the *correctness* of the results and *applicability* for multiple domains. While the observations are generally of *practical benefit*, another important aspect is the *explainability* of the observations. Our observations apply to sensor-based IoT settings where all the data came from IoT sensors. The question is whether or not we can utilise the observations in a root cause analysis.

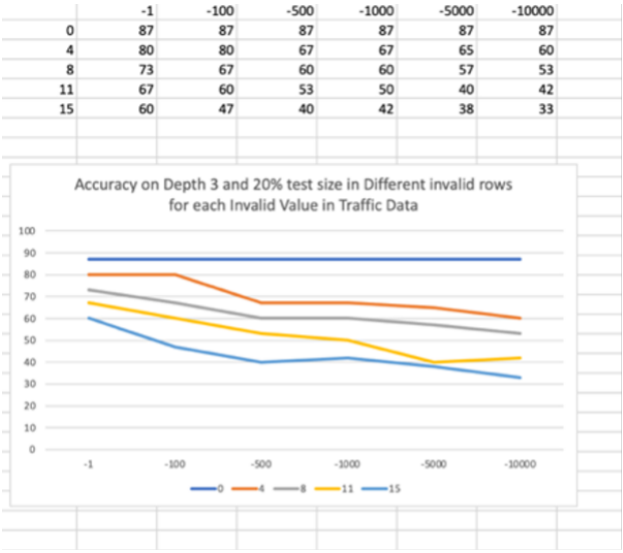

Fig. 1. Accuracy on Depth 3 and 20% Test Size in Different Invalid Rows for Each Invalid Value in Traffic Data.

The missing or invalid rows situation is more likely to happen in real-life situations than missing or invalid features. Data is received from sensors. If a sensor is faulty or the data is not received due to a connection problem, all the data from that sensor is lost (and not a part of data), unless we have different sensors for different factors. In the latter case, it would be possible to have missing features. For example, if a weather sensor can calculate different factors like temperature, humidity, pressure, wind and etc., then if the sensor is faulty, we will lose all the measurement at the same time. If we have different sensors for each measurement, then if the sensor is faulty, we will lose only some at the same time, but not all of them. For invalid values, it depends on the type of sensor and factors. For instance, -50C is generally unlikely for a temperature reading, but still possible to happen; on the other hand below -100C can be assumed incorrect. These observation can be used to deduce probable root causes in sensor-based IoT environments such as faulty sensors or incorrect data processing.

6.1. *Transferability*

We looked at IoT settings, based on sensors as data producers. In that context, we have used traffic and weather data sets.

Other application domains could here be considered, such as mobile learning that includes the usage of multimedia content being delivered to mobile learners and their devices [18, 21, 14]. Here the setting is different in that multimedia content is produced and transferred. This can equally cause incompleteness and incorrectness problems, but here the differences is that continuous streams of binary data is affected.

A further direction is the implementation of self-adaptive ML quality management in an IoT-edge continuum [16, 5, 10, 19]. In that context, similar to the original setting, sensors produce data (albeit sensors measuring often virtualized infrastructure performance) that is

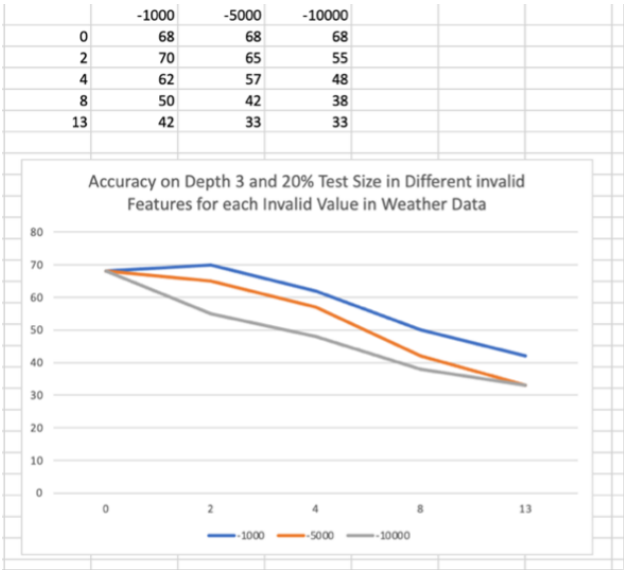

Fig. 2. Accuracy on Depth 3 and 20% Test Size in Different Invalid Features for Each Invalid Value in Weather Data.

after some analysed translated into instructions to be executed by actuators in the same IoT edge space. Here, video surveillance could be considered, where cameras monitor for example building that need protection. Image quality could be monitored and faults in the video or transmission system detected. Microservices and containers [29, 30, 23, 25] would here be the main artifacts that would be monitored, causing continuously produced input data.

From our results, we can ascertain that sensor faults have a different impact than for instance network failures and that with some certainty a defect cause can be identified. This, however, needs to be further explored and confirmed for other data than the numerical and limited volume situations considered here.

7. Conclusion

More and more software applications are based on functions generated using ML from larger volumes of data available in contexts such as the Internet-of-Things (IoT) instead of being manually programmed [24]. With less human involvement in the construction process of the software, quality assurance becomes more important.

We focused on the link between input data quality for ML function construction and the quality of these functions in data-driven software applications. An important observation is the range of quality concerns that apply. For input data, we considered correctness and completeness as data quality concerns. For ML model construction, the usual accuracy, precision and recall were considered. We organized our work in three steps. In first step, we determined a framework of indicators that influence data quality such as correctness and completeness and model construction factors on accuracy, precision and recall as described above. Then, we experimentally analysed the impact of variations in model construction

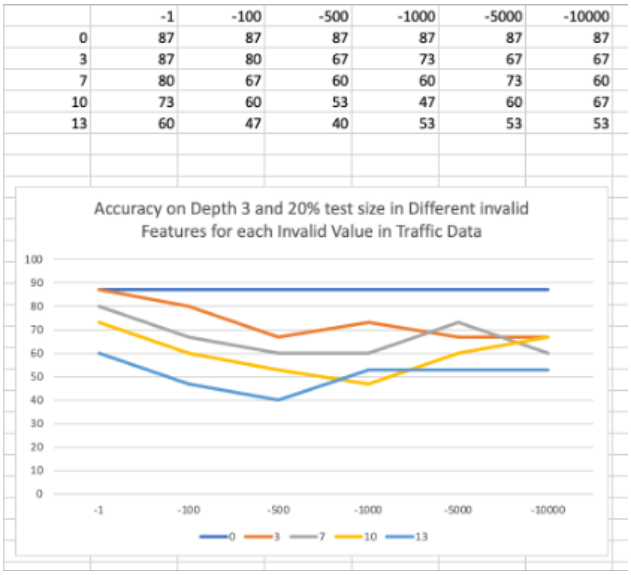

Fig. 3. Accuracy on Depth 3 and 20% Test Size in Different Invalid Features for Each Invalid Value in Traffic Data.

and data quality on ML model quality and in the final step, we aimed to identify change patterns that can be attributed to specific input changes caused by for instance faults in the environment in the context of a root cause analysis. This provides an a-posteriori explanation for a black box explainability setting.

The observations were validated in two data sets and are practically applicable in machine learning quality analysis and root cause analysis. However, a clear identification of the reason behind the observation is not always possible. More work on the white-box explainability of results is needed.

Acknowledgements

This work has been performed partly within a Ph.D. Programme funded through a bursary by the Südtiroler Informatik AG (SIAG).

1. Azimi, S., Pahl, C.: A layered quality framework in machine learning driven data and information models. In: ICEIS (2020)
2. Azimi, S., Pahl, C.: Root cause analysis and remediation for quality and value improvement in machine learning driven information models. In: ICEIS (2020)
3. Azimi, S., Pahl, C.: Continuous data quality management for machine learning based data-as-a-service architectures. In: CLOSER (2021)
4. Azimi, S., Pahl, C.: The effect of IoT data completeness and correctness on explainable machine learning models. In: Strauss, C., Kotsis, G., Tjoa, A.M., Khalil, I. (eds.) Database and Expert Systems Applications - 32nd International Conference, DEXA 2021, Virtual Event, September 27-30, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12924, pp. 151–160. Springer (2021)
5. Barzegar, H.R., Ioini, N.E., Le, V.T., Pahl, C.: Wireless network evolution towards service conti-

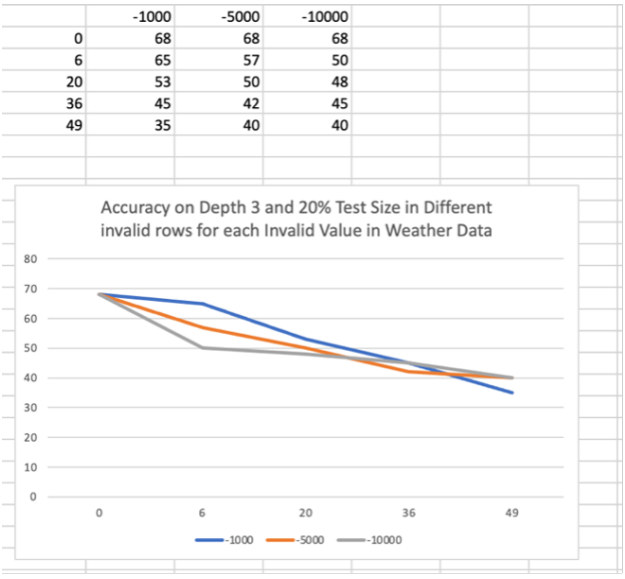

Fig. 4. Accuracy on Depth 3 and 20% Test Size in Different Invalid Rows for Each Invalid Value in Weather Data.

nuity in 5g enabled mobile edge computing. In: Fifth International Conference on Fog and Mobile Edge Computing, FMEC 2020, Paris, France, April 20-23, 2020. pp. 78–85. IEEE (2020)
6. Bosch, J., Olsson, H.H., Crnkovic, I.: Engineering ai systems: A research agenda. In: Artificial Intelligence Paradigms for Smart Cyber-Physical Systems, pp. 1–19. IGI Global (2021)
7. Ehrlinger, L., Haunschmid, V., Palazzini, D., Lettner, C.: A daql to monitor data quality in machine learning applications. In: Database and Expert Systems Applications (2019)
8. Fang, D., Liu, X., Romdhani, I., Jamshidi, P., Pahl, C.: An agility-oriented and fuzziness-embedded semantic model for collaborative cloud service search, retrieval and recommendation. Future Generation Computer Systems **56**, 11–26 (2016)
9. Fowley, F., Pahl, C., Jamshidi, P., Fang, D., Liu, X.: A classification and comparison framework for cloud service brokerage architectures. IEEE Transactions on Cloud Computing **6**(2), 358–371 (2016)
10. Gand, F., Fronza, I., Ioini, N.E., Barzegar, H.R., Le, V.T., Pahl, C.: A lightweight virtualisation platform for cooperative, connected and automated mobility. In: Berns, K., Helfert, M., Gusikhin, O. (eds.) Proceedings of the 6th International Conference on Vehicle Technology and Intelligent Transport Systems, VEHITS 2020, Prague, Czech Republic, May 2-4, 2020. pp. 211–220 (2020)
11. Harp, S., Goldman, R., Samad, T.: Imputation of missing data using machine learning techniques. pp. 140–145 (01 1996)
12. Javed, M., Abgaz, Y.M., Pahl, C.: Ontology change management and identification of change patterns. J. Data Semant. **2**(2-3), 119–143 (2013)
13. Jiarpakdee, J., Tantithamthavorn, C., Dam, H.K., Grundy, J.: An empirical study of model-agnostic techniques for defect prediction models. IEEE Transactions on Software Engineering pp. 1–1 (2020)
14. Kenny, C., Pahl, C.: Automated tutoring for a database skills training environment. In: Proceedings of the 36th SIGCSE Technical Symposium on Computer Science Education. p. 58–62. SIGCSE '05, Association for Computing Machinery, New York, NY, USA (2005)
15. Khosravi, P., Vergari, A., Choi, Y., Liang, Y., Broeck, G.: Handling missing data in decision trees: A probabilistic approach (06 2020)

16. von Leon, D., Miori, L., Sanin, J., Ioini, N.E., Helmer, S., Pahl, C.: A lightweight container middleware for edge cloud architectures. In: Buyya, R., Srirama, S.N. (eds.) Fog and Edge Computing, pp. 145–170. Wiley (2019)
17. Marev, M.S., Compatangelo, E., Vasconcelos, W.W.: Towards a context-dependent numerical data quality evaluation framework. CoRR **abs/1810.09399** (2018), `http://arxiv.org/abs/1810.09399`
18. Melia, M., Pahl, C.: Constraint-based validation of adaptive e-learning courseware. IEEE Trans. Learn. Technol. **2**(1), 37–49 (2009)
19. Mendonça, N.C., Jamshidi, P., Garlan, D., Pahl, C.: Developing self-adaptive microservice systems: Challenges and directions. IEEE Softw. **38**(2), 70–79 (2021)
20. Mittelstadt, B.D., Russell, C., Wachter, S.: Explaining explanations in AI. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019. ACM (2019)
21. Murray, S., Ryan, J., Pahl, C.: Tool-mediated cognitive apprenticeship approach for a computer engineering course. In: International Conference on Advanced Learning Technologies, ICALT. pp. 2–6. IEEE Computer Society (2003)
22. Nikfalazar, S., Yeh, C.H., Bedingfield, S., Khorshidi, H.: Missing data imputation using decision trees and fuzzy clustering with iterative learning (2020)
23. Pahl, C.: An ontology for software component matching. In: International Conference on Fundamental Approaches to Software Engineering. pp. 6–21. Springer (2003)
24. Pahl, C., Azimi, S.: Constructing dependable data-driven software with machine learning. In: IEEE Software (2021)
25. Pahl, C., Jamshidi, P., Zimmermann, O.: Microservices and containers. Software Engineering 2020 (2020)
26. Patel, D., Nguyen, L.M., Rangamani, A., Shrivastava, S., Kalagnanam, J.: Chief: A change pattern based interpretable failure analyzer. In: Intl Conf on Big Data. pp. 1978–1985 (2018)
27. Roscher, R., Bohn, B., Duarte, M.F., Garcke, J.: Explainable machine learning for scientific insights and discoveries. IEEE Access **8** (2020)
28. Shrivastava, S., Patel, D., Zhou, N., Iyengar, A., Bhamidipaty, A.: Dqlearn : A toolkit for structured data quality learning. In: Intl Conf on Big Data. pp. 1644–1653 (2020)
29. Taibi, D., Lenarduzzi, V., Pahl, C.: Continuous architecting with microservices and devops: A systematic mapping study. In: International Conference on Cloud Computing and Services Science. pp. 126–151. Springer (2018)
30. Taibi, D., Lenarduzzi, V., Pahl, C., Janes, A.: Microservices in agile software development: a workshop-based study into issues, advantages, and disadvantages. In: Proceedings of the XP2017 Scientific Workshops. pp. 1–5 (2017)
31. Tantithamthavorn, C., Jiarpakdee, J., Grundy, J.: Explainable ai for software engineering. arXiv preprint arXiv:2012.01614 (2020)
32. Tsang, S., Kao, B., Yip, K., Ho, W.s., Lee, S.: Decision trees for uncertain data. In: Proceedings International Conference on Data Engineering (2009)
33. Twala, B., Jones, M., Hand, D.: Good methods for coping with missing data in decision trees. Pattern Recognition Letters **29**, 950–956 (05 2008)