

The Effect of IoT Data Completeness and Correctness on Explainable Machine Learning Models

Shelernaz Azimi and Claus Pahl

Free University of Bozen-Bolzano, Bolzano, Italy
{fname.sname}@unibz.it

Abstract. 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.

Keywords: Explainable AI, 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 [11]. 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.

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.

2 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 [4], [13], [7], [18]. 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. [13] addresses missing data imputation. The authors propose a method called DIFC integrating decision trees 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 [5]. [19] also proposed a method for dealing with missing data in decision trees. In [7], 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. [18] 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 [15] 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 [17], 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.

3 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. 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 [6]. 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.

Our models here are decision trees – using scikit-learn¹ to both data sets for predictions. Using traffic data, 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. The second data set 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. The data from both data sets consisting of only numerical values has been processed and labeled manually.

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

¹ <https://scikit-learn.org/> - Machine learning library for Python

Table 1: 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.

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.**

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 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.

Table 2: 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.

4 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. In total, we conducted more than 50 experiments that varied settings in 4 dimensions (tree depth, test size, missing/invalid features, missing/invalid rows), which cannot be presented here in full. 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.

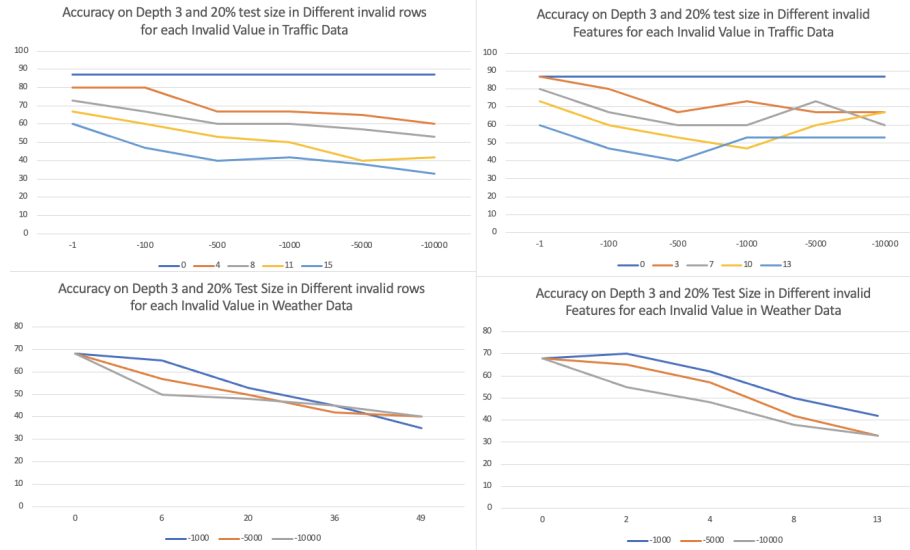


Fig. 1: Experimental results for the effect of different invalid values on both data sets for invalid rows and invalid features.

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 applicable 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.

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 [16]. 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.

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.

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.

5 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 [14]. 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 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 a post-hoc 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. 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 [9, 12]. A further direction is the implementation of self-adaptive ML quality management in an IoT-edge continuum [8, 10].

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

References

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. 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)
5. Harp, S., Goldman, R., Samad, T.: Imputation of missing data using machine learning techniques. pp. 140–145 (01 1996)
6. Javed, M., Abgaz, Y.M., Pahl, C.: Ontology change management and identification of change patterns. *J. Data Semant.* **2**(2-3), 119–143 (2013)
7. Khosravi, P., Vergari, A., Choi, Y., Liang, Y., Broeck, G.: Handling missing data in decision trees: A probabilistic approach (06 2020)
8. 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)
9. Melia, M., Pahl, C.: Constraint-based validation of adaptive e-learning courseware. *IEEE Trans. Learn. Technol.* **2**(1), 37–49 (2009)
10. Mendonça, N.C., Jamshidi, P., Garlan, D., Pahl, C.: Developing self-adaptive microservice systems: Challenges and directions. *IEEE Softw.* **38**(2), 70–79 (2021)
11. 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)
12. 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)
13. Nikfalazar, S., Yeh, C.H., Bedingfield, S., Khorshidi, H.: Missing data imputation using decision trees and fuzzy clustering with iterative learning (2020)
14. Pahl, C., Azimi, S.: Constructing dependable data-driven software with machine learning. In: *IEEE Software* (2021)
15. 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)
16. Roscher, R., Bohn, B., Duarte, M.F., Garcke, J.: Explainable machine learning for scientific insights and discoveries. *IEEE Access* **8** (2020)
17. 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)
18. Tsang, S., Kao, B., Yip, K., Ho, W.s., Lee, S.: Decision trees for uncertain data. In: *Proceedings - International Conference on Data Engineering* (2009)
19. Twala, B., Jones, M., Hand, D.: Good methods for coping with missing data in decision trees. *Pattern Recognition Letters* **29**, 950–956 (05 2008)