Improving Machine Learning-based Decision-Making through Inclusion of Data Quality

Jens de Hoog ✉, Siegfried Mercelis, and Peter Hellinckx

University of Antwerp - imec
IDLab - Faculty of Applied Engineering
Sint-Pietersvliet 7, 2000 Antwerp, Belgium
{jens.dehoog, siegfried.mercelis, peter.hellinckx}@uantwerpen.be

Abstract. Nowadays, numerous Internet of Things devices are producing large amounts of data. This data originates from the environment in which these devices are operating. In theory, these devices sample the environment in a way which is closest to reality, but in practice, this is far from ideal. On the one hand, this is due to imperfections in the sensory devices. On the other hand, this could be due to a complex or volatile environment, which is difficult to perceive for such sensors. This problem can be solved by adding external environmental data to simplify the perception, but the quality of that external data is unknown. In both cases, there is a degree of uncertainty in the data. In this paper, we introduce a concept in which the quality of data is measured and is incorporated in a particular Decision-Making Process based on machine learning paradigms. In this way, the decisions are made with knowledge about that quality. We position this concept in the current State-of-the-Art regarding data quality and machine learning architectures. Additionally, this paper elaborates on a hypothetical example using the proposed concepts.

Keywords: data · quality · machine learning · decision process

1 Introduction

These days, a lot of data is being generated by numerous interconnected Internet of Things (IoT) devices. Many different decision-making processes (DPs) rely on this data to trigger events or control actuators. This can range from climate control applications and fall detectors to industrial processes and lane keeping assistants in autonomous driving. However, sometimes these decisions can go wrong. For example, failures in autonomous vehicles such as Uber or Tesla, and fall detectors that detect false positives or, even worse, false negatives. The cause of these problems is sometimes due to bad input data, thus bad decisions are made. In other cases, the input data is technically correct, but the environment is too complex, difficult or volatile to perceive for sensors. Therefore, by adding external environmental data, this environment is simplified. However, in such an environment the quality of this data is unknown and can be variable over time due to a changing environment. An example is a distributed environment in which multiple heterogeneous entities are able to sense, act and communicate. By extension, the root cause of these problems is the difference between algorithms trained

Copyright © 2019 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR-WS.org/Vol-2491/short14.pdf

in a controlled environment, and algorithms deployed in an environment out of our control.

We can narrow these problems down to the lack of knowledge about the data quality. Within this research, we aim to incorporate this data quality into the DP, so that decisions are based on that quality. An example of such a DP is a machine learning network. Based on several data inputs, it generates a certain output or decision. This incorporation of data quality in such a machine learning process introduces several challenges. First, an abstract and objective measure of data quality needs to be defined. Second, this quality field should be introduced to the machine learning network in a viable way, along with the data itself. Additionally, based on the data and its quality, which can vary over time, proper decisions should be generated accordingly.

In this paper, we position our vision in the current State-of-the-Art regarding data quality and the inclusion of it in a DP based on machine learning. This paper is structured as follows. Section 2 elaborates on data quality; that is, different measuring techniques as well as representations for data quality are discussed. The section concludes with the positioning of our research in this field. Section 3 discusses several data fusion techniques in a machine learning architecture, followed by the positioning of our research. Section 4 enlightens several approaches concerning the incorporation of data quality in a machine learning network. Section 5 elaborates on a hypothetical example in which the four conceptual approaches are enlightened Finally, section 6 concludes this paper.

2 Measuring Data Quality

The quality of data can be measured in various ways. First, a context can be created on this data. According to Kim et al. [8], a context can be seen as information about a certain situation at the time of a particular interaction. Such information includes raw sensor data, data of a user interaction, data produced by the application itself, etc. The quality of this data can be evaluated based on different Key Performance Indicators or KPIs, such as the accuracy of the data (evaluated by an observed data value and its expected value), the representation consistency and completeness. The quality of the created context is also quantifiable; this is called Quality of Context (QoC). Based on the research of Kim et al. [8] and Al-Shargabi et al. [1], this QoC can be measured by several KPIs, which are rather similar to the aforementioned KPIs. Examples are timeliness, reliability, resolution, probability of correctness and completeness. Every KPI for measuring the QoC is calculable. For example, the timeliness determines the freshness of the context; i.e., the age of the created context is compared with the prescribed time span in which the context is valid. This implies that a more fresh context results in a higher quality. For example, context information that is received a few seconds ago, is closer to reality than information of the previous day, hence the higher quality. It is clear that such a Quality of Context provides an objective measure on the quality of data.
Additionally, research has been conducted in providing a quality measure on the content of the data itself. For example, Berkvens [2] conducted a study regarding Information Theory. He points out that the conditional entropy of information and the mutual information are useful metrics when measuring respectively the uncertainty and certainty

of information content. Finally, Karkouch et al. [7] propose a system in which the data quality of IoT data is enhanced. They do so by first analysing the quality itself (based on approximately the same KPIs as proposed by [1] and [8]), after which they enhance the data by several approaches such as data interpolation, deduplication, etc.

In terms of the representation of a quality measure, several in-depth studies have been conducted. Laranjeiro et al. conducted a survey about data quality [9]. They discussed the problems that come with low quality data, different techniques to measure that quality using prescribed metrics, all from several perspectives such as enterprises, end users and researchers. The authors represented the overall quality of data via different quality dimensions. Examples are completeness, accuracy, consistency, etc. These dimensions correspond to the KPIs mentioned in previous subsection.
Cichy et al. also did a comprehensive survey on data quality, which continues on the work of Laranjeiro [6]. In this survey, the authors address several frameworks for assessing data quality. The authors also state that the overall data quality can be represented via relevant dimensions, which correspond to the aforementioned dimensions of [9].
It is clear that the overall data quality is not represented as a single value, but rather in several quality dimensions. Each dimension assesses another perspective of the data quality, resulting in a more accurate and complete representation of the quality of data.

In terms of our research, all of the above-mentioned contributions are helpful when measuring data quality. Each of these works enlightens another aspect of data and its quality, going from an abstract context measurement to in-depth analysis of the content. Hence, all of these measurement techniques are useful to obtain clear insights about the data, thus being advantageous for the subsequent DP.

3 Including data quality in machine learning networks

To incorporate data quality in a machine learning process, the quality should be fed into the network. Subsequently, by fusing the quality information with the actual data, the incorporation is achieved. As the quality information can also be considered as data, this fusion process is called *data fusion*.

Different techniques exist to perform data fusion. First, Chen et al. propose a method to improve prediction accuracy for traffic flow predictions, based on spatial-temporal data with uncertainty [4]. They do so by integrating fuzzy logic with deep learning techniques. The network is split in two parallel channels: a fuzzy network and a deep convolutional network. The data is fed to the input, after which it is processed by the two channels in parallel. The outputs of both channels are then fused together and a prediction is made by the last layer.
Patel et al. proposed a neural network architecture called *NetGated architecture* [10]. This network performs data fusion for a LiDAR sensor and a camera, after which it outputs a steering command for the robot of the authors. The fusion process starts with a deep convolutional neural network for both sensors. Each network outputs a feature vector for that sensor. The vectors are concatenated and processed by a fully connected layer, after which a weight for each feature vector is generated. These weights are multiplied with the same feature vectors as before (i.e. the outputs from the convolutional neural networks). Afterwards, these weighted vectors are processed by two fully con-

nected layers. The fact that weights are generated from these feature vectors and are multiplied with them, causes this structure to behave as a gate for each vector. In this way, the influence of each sensor on the global output can be changed adaptively.
Shim and Li propose two optimised variants of the aforementioned *NetGated* network [11]. They address the problems that the original version has: (i) the network has an increased potential of overfitting during the training process, and (ii) the fusion weights sometimes do not correspond to its originating feature vector as it contains information about both vectors. The first proposed variant deals with grouping multiple sensors and their convolutional outputs together to calculate a single weight on that group. Their structure has five inputs; these inputs are split into two groups of respectively three and two inputs. After the calculation of both weights, these are multiplied with their corresponding group, after which an output can be generated in the same way as in the original architecture. This approach minimises the overfitting potential of the network. Additionally, the weights are now more likely to correspond to its group than they did with its feature in the original version. This is also the downside of the architecture at the same time; the weight does not contain information about a particular feature. Therefore, the authors propose a second construction that is based on both the original one and their first proposed network. That is, the feature weights are calculated for each separated input, along with weights calculated on groups of inputs. These weights of both features and groups are then multiplied with each other, which results in new weights containing information of both features and groups. This architecture solves both problems of the original one and shows significant improvements.
Other studies deal with uncertain data in machine learning networks during the training phases, resulting in robust networks that are able to process uncertain data during evaluation. For example Choi et al. conducted a comprehensive study concerning the training process of neural networks with noisy data and labels [5]. Their proposed network, called ChoiceNet, is able to learn noisy and corrupt datasets of images (i.e. MNIST and CIFAR-10) and outperforms the existing networks regarding training accuracy.

The above-mentioned State-of-the-Art provides clear insights and ideas for incorporating data quality in a machine learning network. First, the data quality measures can be fed into the network as a conventional input. In this way, the network learns itself to deal with these quality measures. Robust networks, such as ChoiceNet from Choi et al., could benefit from these quality measures in terms of accuracy and robustness during both training and evaluation phases. Another option is to make use of the aforementioned *NetGated* networks by [10] and [11]. However, other architectures for processing the data quality would also be appropriate instead of being limited to a convolutional neural network. In this way, we can extract abstract features out of these quality measures, which can contribute to the fusion weights. Thus, these quality measures or their abstract features would serve as gate for the feature vectors of the other inputs. Finally, the work of [4] also proposes a possible approach for the inclusion of data quality. If these quality measures could act upon the inner logic inside the fuzzy network, this quality would then have an impact on the fusion process with the deep learning network and the global output would change accordingly.

4 Conceptual approaches

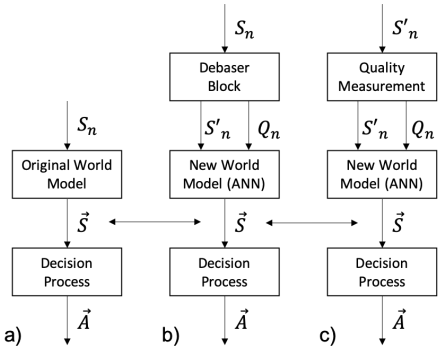

Fig. 1. Conceptual approach 1

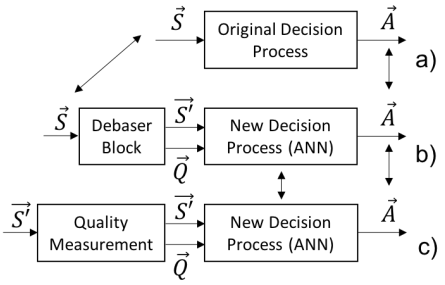

Fig. 2. Conceptual approach 2

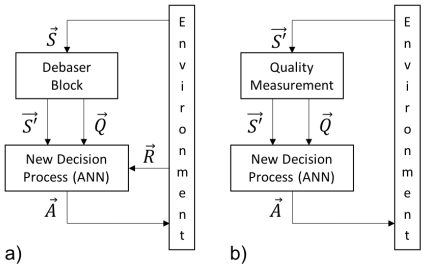

Fig. 3. Conceptual approach 3

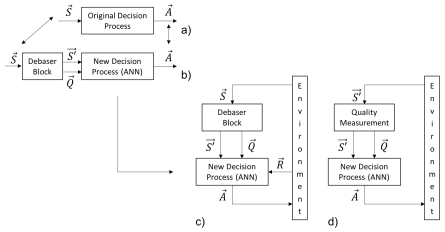

Fig. 4. Conceptual approach 4

Given the aforementioned techniques concerning machine learning networks and data quality, we consider four possible approaches that would contribute as improvements to the current State-of-the-Art.

To start, we opt to use a system in which we will incorporate data with its quality labels, but with an already existing DP. In this way, the DP is fully dependent of the design choices of the overall system. Consider a system in which no quality labels are taken into account; such a system is shown in Figure 1a. We introduce a World Model (WM), which is the internal state of the system; this state will change according to the incoming data. Therefore, based on this internal state, the DP will calculate a decision. If we now consider a system in which debased data (i.e. decreased quality due to low resolution, inconsistency, noise, etc.) and its quality labels are taken into account, the DP shall not be changed. This means that these quality labels need to be incorporated into this WM, such that it generates an appropriate state based on those labels. That is, the WM will shape its output in such a way that the low quality data is treated differently than if the data was not accompanied by a quality label. As this results in a different behaviour of the WM, the behaviour of the DP will change indirectly. The recreation of

the WM can be accomplished by using one of the aforementioned techniques in section 3. Though, ChoiceNet from Choi et al. or a similar network is preferred as it can recreate an output based on noisy input data, thus ideal for the recreation of our World Model. The training procedure is as set up as follows. First, we use incoming data which is considered perfect. By feeding this data into an existing world model, we can collect the states this model generates (shown in Figure 1a). We consider these states as the baselines. Next, the same incoming data is used, but its quality is degraded by a known amount; hence, a quality label can be generated accordingly. Now, the input and output of the neural network in the training phase are respectively the debased data with its quality label and the state generated by the original world model. Therefore, the newly made WM will try to fit as close as possible to the original neural network. When the training process has finished and the model has proven its feasibility, the next stage is to use a quality measurement module for imperfect incoming data. This stage is shown in Figure 1c. A potential issue with this approach is the lack of generalization on different kinds of data imperfections. As long as the imperfections are similar to the ones learned during the training phase, the network should be able to generate an appropriate output. However, if there are imperfections present that are not learned in the training phase, the generated output could be inaccurate.

The first approach creates a clear view of the possibilities regarding the incorporation of data quality into the DP, but the methodology has a major flaw. The WM constantly needs to recreate a perfect state as if it was created with perfect incoming data instead of debased data; the independent DP is only capable of calculating actions based on this perfect state. Hence, this regeneration creates an unnecessary complexity in the architecture. To overcome this complexity, we will add the DP into the learning process such that the WM and DP have an internal link with each other instead of being two separate units. Figure 2 shows a schematic of this approach. In terms of implementation, any of the techniques mentioned in section 3 is applicable. That is, if ChoiceNet or similar is used, the same decisions would be generated as if perfect data was fed to the network. Additionally, the NetGated architecture also can draw decisions based on the quality measures and has the ability to interpret the resulting feature vectors. Therefore, this network can draw decisions that are the most suitable for a given situation. However, this feature is out of scope for this approach. The training procedure is similar to the previous approach. That is, we start from incoming data which is considered perfect. Based on this data, we are able to collect the action generated by original DP (shown in Figure 2a). The next stage is shown in Figure 2b, which is similar to the one shown in Figure 1b. The final step, which is shown in Figure 2c, incorporates a quality measurement unit. This stage is also similar to the one shown in Figure 1c.

As mentioned before, the first approach establishes a clear view about the possibilities of incorporating quality measures of the incoming data streams. The second approach resolves a major flaw of the first one: the system does not need to explicitly regenerate a perfect state that is required for the DP to operate. The World Model and DP are now one entity with the same behaviour as the two separate entities of the first approach. The problem in the second approach is that this behaviour is limited to the behaviour of the original DP. This means that, if this original one has weaknesses in the process of calculating an action, the newly generated DP has these same weaknesses.

Those are propagated into the newly created DP as the actions of the original DP are used as baseline. Thus, the new DP is limited to the actions of the original one while sharing the same flaws. This problem will be solved by using a Reinforcement Learning technique. The DP will learn itself to calculate the most appropriate action for a given data stream of a certain situation. Figure 3a shows an environment which generates perfect data, after which this data is being debased; this is similar to the previous approaches. The generated action is given back to the environment, resulting in a corresponding reward for that action. This reward is given back to the DP, so it takes this feedback into account when generating a new action. When the learning procedure of the DP has been successful, the DP is now used in the same way as in the previous approach. This is shown in Figure 3b. In contrast to the aforementioned approaches, ChoiceNet or similar is less applicable for this situation. That is, the network has to draw the most suitable decision for the given situation instead of recreating an output based on noisy input data. Therefore, the other techniques mentioned in section 3 are more suitable.

In the previous approach, the DP has the freedom to learn itself the appropriate actions for a particular situation. The advantage of that approach is that the DP is not dependent of an already existing one, which may contain some weaknesses. The disadvantage of that approach is the gigantic space in which the DP needs to find an optimal model, due to the large amount of combinations of different data streams and their quality measures. This makes the training process much slower, which could be a bottleneck when designing a DP for a particular application. For example, a DP specifically designed for an autonomous vehicle is not usable in an application designed for small IoT-devices such as temperature sensors. Therefore, a new DP needs to be trained specifically for that application, which could take a considerable amount of time. To overcome this problem, we believe a hybrid approach provides the best of both worlds. More specifically, we will create a DP based on Supervised Learning (as proposed in the second approach, shown in Figure 4a and 4b), after which we further improve this DP via Reinforcement Learning (as proposed in the third approach, shown in Figure 4c). This means that our DP could be based on an already existing application-specific DP, but it would not be limited to the original one due to the improvements via Reinforcement Learning. We also think that the time required to train the model is longer than required in the second approach, but shorter than in the third approach due to the head start of the model. Hence, we think this approach solves the drawbacks of the previous ones, resulting in an overall better approach (shown in Figure 4d). In terms of implementation, the same machine learning network from the previous approach would be used as the decisions need to be the most suitable for the given situation.

5 Hypothetical Example

In the previous section, this paper proposed four methodologies to incorporate data quality into a DP. In this section, we form a hypothetical example in which we use fictional sensor data accompanied by its quality indicator, applied on the four aforementioned approaches. Consider an office room in which multiple temperature sensors are positioned over the entire room. An example of such a room is shown in Figure 5,

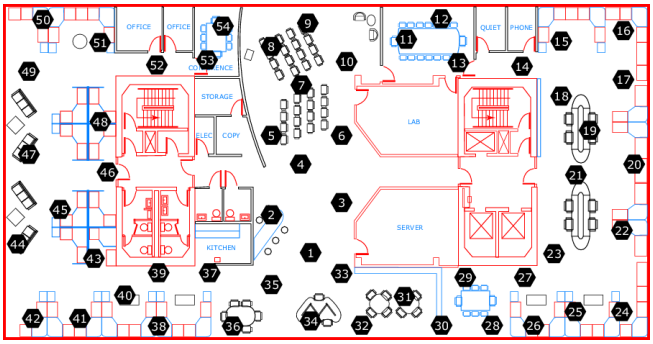

Fig. 5. Office room of Intel Berkeley Research Lab

which is the Intel Berkeley Research Lab [3]. A fictional central heating system creates a heat map of the entire room, after which the level of each separate heating unit is adjusted to accomplish a comfortable temperature in the office environment. Consider a fictional temperature sensor whose updates to the central heating system are lagging in time by a couple of hours. This means that the data generated by the sensor is not accurate with respect to the current temperature of the environment. Therefore, we can determine the quality of the sensor data with respect to the timeliness of the data, which is low.

In the first approach mentioned in section 4, the WM will take the quality into account. In this example, the WM can be represented as the heat map of the office room. Based on this model, the DP adjusts its heating units. As the quality (i.e. the timeliness) of the aforementioned fictional sensor is low, the WM needs to be aware of this low quality via the given quality labels; it will treat this low quality data in a different way. In this example, the WM will not take this sensor data as much into account as it does with other sensors to generate the heat map of the environment; the other sensors are considered perfect. Hence, the central heating system is not aware of the low quality sensor data, but it will behave different nevertheless due to the adapted WM.

In the second conceptual approach, the DP itself will incorporate the quality labels such that it bases the decisions on those labels. In this example, the DP is based on an existing central heating process, e.g. a rule-based system. On the one side, the created neural network will then behave in the same way as the original one does, with the added ability of taking the quality labels into account. Hence, the low quality sensor data will not have as much impact on the decision as the high quality sensors have. This means that the DP will only focus on the sensors that represent the current temperature instead of also taking the faulty data into account. On the other side, as mentioned in the previous section, the neural network will be limited to the behaviour of the original process, which leads to inheriting the same weaknesses as the original one.

In the third conceptual approach, the DP is created from scratch using Reinforcement Learning. This means that the central heating process is not limited to an original process any more, which leads to a possibly more optimised solution for the office environment, while also taking the quality of its sensor data into account. Therefore, the

network would be able to take other, possibly more optimised actions if the timeliness of a particular sensor is low. As mentioned in the previous section, the downside of this approach is that it can take a long time to reach that optimal solution during the training stage.

Therefore, regarding the fourth conceptual approach, the newly created network is first based on an existing heating process, after which it is optimised for the specific office environment. This results in a head start regarding the Reinforcement Learning technique. The outcome of this fourth approach is a heating process that is able to behave in a more optimised way than an existing heating process regarding the office environment, along with the ability to take a appropriate actions when the quality of a temperature sensor is low.

6 Conclusion

In this paper, we have proposed a new concept of incorporating data quality in a Decision-Making Process (DP), which is based on machine learning paradigms. Doing so, the DP would make more appropriate decisions based on the given quality of its inputs. We have enlightened the current State-of-the-Art regarding data quality and data fusion in machine learning, after which we positioned our concept into these topics. We proposed four approaches to achieve this inclusion of data quality. First, we include the quality into the world model on which the DP bases its decisions, but in the second approach we include the quality into the DP itself. In the third approach, we propose the usage of a DP based on Reinforcement Learning techniques. Finally, in the fourth approach, we propose a hybrid architecture in which we include the quality in a DP and improve this via Reinforcement Learning. We believe that this last approach is the most flexible one as it starts from an existing process, but is further improved later on to gain overall robustness for a broader range of use cases. This paper elaborated on a hypothetical example focused on a specific IoT application. However, more research is needed to provide a more thorough answer on our conceptual statements, along with an elaboration on the generalization for other application domains, different types of data and their possible quality measurements.

References

1. Al-Shargabi, A.A., Siewe, F., Zahary, A.T.: Quality of Context in Context-Aware Systems. EAI Endorsed Transactions on Context-aware Systems and Applications **4**(12) (2017). https://doi.org/10.4108/eai.6-7-2017.152761, http://eudl.eu/doi/10.4108/eai.6-7-2017.152761
2. Berkvens, R.: Uncertainty of localization using electromagnetic fingerprints. Ph.D. thesis, University of Antwerp (2017)
3. Bodik, P., Hong, W., Guestrin, C., Madden, S., Paskin, M., Thibaux, R.: Intel lab data. Online dataset (2004)
4. Chen, W., An, J., Li, R., Fu, L., Xie, G., Bhuiyan, M.Z.A., Li, K.: A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial-temporal data features. Future Generation Computer Systems **89**, 78–88 (2018). https://doi.org/10.1016/j.future.2018.06.021

5. Choi, S., Hong, S., Lim, S.: Choicenet: Robust learning by revealing output correlations. CoRR **abs/1805.06431** (2018), http://arxiv.org/abs/1805.06431
6. Cichy, C., Rass, S.: An Overview of Data Quality Frameworks. IEEE Access **PP**(c), 1–1 (2019). https://doi.org/10.1109/ACCESS.2019.2899751, https://ieeexplore.ieee.org/document/8642813/
7. Karkouch, A., Mousannif, H., Al Moatassime, H., Noel, T.: Data quality in internet of things: A state-of-the-art survey. Journal of Network and Computer Applications **73**, 57–81 (2016). https://doi.org/10.1016/j.jnca.2016.08.002, http://dx.doi.org/10.1016/j.jnca.2016.08.002
8. Kim, Y., Lee, K.: A Quality Measurement Method of Context Information in Ubiquitous Environments. 2006 International Conference on Hybrid Information Technology **2**, 576–581 (2006). https://doi.org/10.1109/ICHIT.2006.253664, http://ieeexplore.ieee.org/xpl/freeabs{_}all.jsp?arnumber=4021269
9. Laranjeiro, N., Soydemir, S.N., Bernardino, J.: A Survey on Data Quality: Classifying Poor Data. Proceedings - 2015 IEEE 21st Pacific Rim International Symposium on Dependable Computing, PRDC 2015 pp. 179–188 (2016). https://doi.org/10.1109/PRDC.2015.41
10. Patel, N., Choromanska, A., Krishnamurthy, P., Khorrami, F.: Sensor modality fusion with cnns for ugv autonomous driving in indoor environments. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 1531–1536 (Sep 2017). https://doi.org/10.1109/IROS.2017.8205958
11. Shim, M.S., Li., P.: Optimized gated deep learning architectures for sensor fusion. CoRR **abs/1810.04160** (2018)