# Uncertainty Wrappers for Data-driven Models

## Increase the Transparency of AI/ML-based Models through Enrichment with Dependable Situation-aware Uncertainty Estimates

Michael Kläs[1] and Lena Sembach[1]

[1] Fraunhofer Institute for Experimental Software Engineering IESE,
Fraunhofer Platz 1, 67663 Kaiserslautern, Germany
{michael.klaes, lena.sembach}@iese.fraunhofer.de

**Abstract.** In contrast to established safety-critical software components, we can neither prove nor assume that the outcomes of components containing models based on artificial intelligence (AI) or machine learning (ML) will be correct in any situation. Thus, uncertainty is an inherent part of decision-making when using the outcomes of data-driven models created by AI/ML algorithms. In order to deal with this – especially in the context of safety-related systems – we need to make uncertainty transparent via dependable statistical statements. This paper introduces both a conceptual model and the related mathematical foundation of an uncertainty wrapper solution for data-driven models. The wrapper enriches existing data-driven models such as provided by ML or other AI techniques with case-individual and sound uncertainty estimates. The task of traffic sign recognition is used to illustrate the approach, which considers uncertainty not only in terms of model fit but also in terms of data quality and scope compliance.

**Keywords:** Artificial Intelligence, Machine Learning, Dependability, Safety Engineering, Data Quality, Operational Design Domain, Model Validation.

## 1 Motivation

More and more software-intensive systems contain components that make use of data-driven models (DDMs) [1], [2], such as those provided by the application of AI and ML. In particular, autonomous systems need to process various kinds of sensor input to recognize and interpret their context and collaborate with other agents. Unlike traditionally engineered components, which are developed by software engineers who define their functional behavior by writing code or models, the behavior of data-driven components (DDCs) is determined by algorithms based on a sample of training data.

The functional behavior expected from DDCs can therefore only be specified and tested on a selection of example cases, and we cannot assure that DDCs will behave as intended in all cases. As a consequence, the outcome of DDCs is afflicted with uncertainty, which has to be appropriately understood and managed during design time and runtime to provide a dependable system.

Previous work [3] proposes separating the sources of uncertainty in DDCs into three major classes, distinguishing between uncertainty caused by limitations in terms of model fit, data quality, and scope compliance. Whereas *model fit* focuses on the inherent uncertainty in a DDM, *data quality* covers the additional uncertainty caused by its application to input data obtained in suboptimal conditions. *Scope compliance* finally covers the phenomenon that a model might be applied in situations outside the scope for which it was intended (i.e., for which it was trained and tested).

Building upon this classification, we derive and illustrate a mathematical model for capturing information about the different sources of uncertainty and combining it into situation-aware and statistically sound uncertainty statements (Section 3). First, Section 2 motivates this work, positioning it in the context of existing work. The paper concludes with possible applications and provides an outlook on future work (Section 4).

## 2 Related Work

Uncertainty estimates are usually expressed by probabilities for categorical outcomes and by probability distributions or prediction intervals in combination with a confidence level for numerical outcomes [4]. In the literature, different approaches can be identified for obtaining such estimates [5].

In the domain of computation and simulation models, uncertain model inputs are commonly addressed by forward uncertainty propagation [6]. Founded on the propagation of error theory, related techniques consider probability distributions instead of concrete values as model inputs to express inherent uncertainty and propagate them through the model. In the context of DDMs, however, it appears difficult to express uncertainty in unstructured data such as images using a probability distribution. Moreover, computation time requirements complicate reasonable applications at runtime.

In the context of AI/ML, some common techniques used to train models implicitly include a kind of uncertainty information (e.g., decision trees [7] also provide probability values for each class, besides their categorical outcomes). For other techniques, revisions have been proposed to provide uncertainty estimates (e.g., for some (deep) neural networks [8], [9] [10], hybrid models [11], and analogy-based techniques [12]). Moreover, some meta-techniques are available that can be applied on top of existing modeling techniques [13] [14] to obtain uncertainty estimates.

A limitation observed for many existing techniques that include means for providing uncertainty estimates is that these estimates are computed on training data. In consequence, they may suffer from overfitting, leading to overconfidence when applied to yet unseen data during application. A further drawback of integrating the calculation of uncertainty estimates directly into a DDM is that the provision of realistic uncertainty estimates and the provision of accurate outcomes do not necessarily require the same inputs and features. For example, the meta-information that an image has low quality does not help to recognize an object in this image, but may indicate that there is a higher degree of uncertainty in the outcome of the applied object recognition model. Finally, existing approaches largely ignore the fact that a DDM might also be applied outside the target application scope for which it was trained and tested.

# 3 Uncertainty Wrapper Approach

In this chapter, we derive and illustrate a mathematical model that allows encapsulating an existing DDM in a DDC via an uncertainty wrapper, which extends the outcome of the DDC ($o^m$) by a situation-aware uncertainty estimate ($\hat{u}^\alpha$) considering a requested degree of confidence (α). Such a DDC can be, e.g., a traffic sign recognition component that decides in a given situation ($case$) based on the provided input ($in$), e.g. an image, whether the detected sign is or is not a 'Stop Sign' (Fig. 1). To provide situation-aware uncertainty estimates, the wrapper addresses model fit, data quality, and scope compliance related uncertainty [3] using the outcome of the DDC, a quality impact model, and a scope compliance model to process the respective parts of the uncertainty estimate.

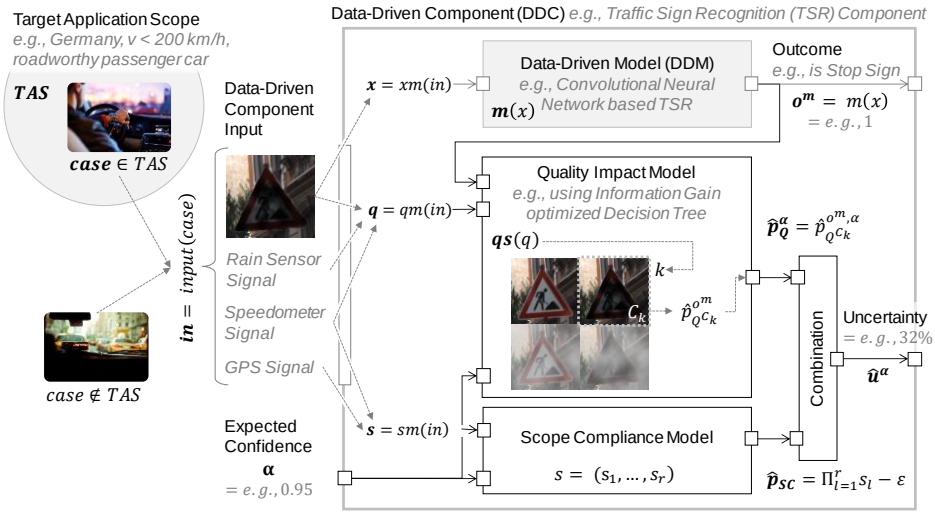


**Fig. 1.** Overview of important concepts and the data flow when providing uncertainty estimates

In the following, we will limit our considerations to binary outcomes (i.e., $o^m \epsilon \{0,1\}$) for the sake of simplicity. However, an extension to DDCs with an arbitrary number of categories as possible outcomes is straightforward (cf. one hot encoding).

In the following, we will first introduce the fundamental mathematics needed to probabilistically describe uncertainty for our setting; next, we will separate uncertainty into a quality-related ($p_Q$) and a scope-compliance-related ($p_{SC}$) part. Then we will propose estimators for both parts and finally combine them into an overall statement on case-specific uncertainty ($\hat{u}^\alpha$).

**Uncertainty.** We define a ***case*** as a specific situation in which the DDC should provide an outcome; for example, passing with a specific car at a certain point in time at a specific location, which leads to an ***input in*** to the DDC. This input can include, for example, the image with the detected traffic sign, which needs to be identified, but also other sensor signals provided, e.g., by the GPS, rain sensor, and speedometer. $in$, which can be assumed as a tensor of a predefined structure, is the result of applying a measurement function $input$ to the case, i.e., $in = input(case)$.

To provide an outcome, parts of $in$ are preprocessed and provided to the DDM with the ***model function m*** generating the outcome, i.e., $o^m = m(x)$ with $x = xm(in)$. In our example, $x$ could be a tensor representing the normalized RGB channels of all pixels of the camera image. Let us now assume that we have some realizations $(x_1, o_1), (x_2, o_2), \ldots$ of random variables $(X, O)$ where $o_i$ is the true label, i.e., the real outcome of a case. $o_i$ is known for training and test cases but not for cases in the application phase, and $o_i^m$ is the outcome by applying $m$ on $x_i$. Accordingly, the random variable $O$ models the true label associated with $X$ and $O^m = m(X)$ the predicted ones.

Let us further assume that the realizations $(x_i, o_i)$ are captured under common conditions $S = \{s_1, \ldots s_N\}$, under which the DDM is intended to be applied later on. This defines the ***target application scope TAS*** of the model. The $TAS$, which is also called operational design domain in the context of autonomous driving, could be traveling on German public roads with a roadworthy passenger car and a max. speed of 200 km/h. For reasons of completeness, we further define $UAS$ as a superset of $TAS$ containing all possible cases, i.e., under intended as well as unintended conditions. $UAS$ is highly abstract since it contains, put simply, "the whole universe".

Now we can define ***uncertainty*** as the probability that DDM outcomes are wrong:

$$u\big((X, O)\big) = p(m(X) \neq O) = 1 - p(m(X) = O) \tag{1}$$

**Separating quality ($p_Q$) and scope compliance ($p_{SC}$).** When dealing with uncertainty during the application, we cannot simply assume that the case for which we make an estimate is part of the target application scope $TAS$. Thus, we apply the law of total probability to separate $cases \in UAS$ into cases that are in $TAS$ and cases that are not[1]:

$$\begin{aligned} p(m(X) = O) = {} & p(case \notin TAS)\, p(m(X) = O \mid case \notin TAS) \\ + {} & \underbrace{p(case \in TAS)}_{=: p_{SC}}\, \underbrace{p(m(X) = O \mid case \in TAS)}_{=: p_Q} \quad \geq p_{SC}\, p_Q \end{aligned} \tag{2}$$

As we want to derive a case-specific estimate of uncertainty in our approach and as $TAS$ might be highly imbalanced (e.g., more '0's than '1's in the case of 'stop signs'), uncertainty estimates should distinguish between cases where $o^m = 0$ and where $o^m = 1$. We also indicate this in Fig. 1 with an arrow from the outcome to the quality impact model. Hence, the considerations below are limited to those cases for which the model predicts a stop sign ($o^m = 1$); the consideration for $o^m = 0$ is analogous. Thus $X$ now denotes more specifically the conditional random variable $X \mid m(X) = 1$.

**Scope compliance ($p_{SC}$).** We assume that for each case some scope-related information is available. Without loss of generality, we assume that the first $r$ characteristics of $S$ are measurable and can be checked on $in$ via the function $sm$, i.e., $s = sm(in) = (\mathbf{1}_{s_1}, \mathbf{1}_{s_2}, \ldots, \mathbf{1}_{s_r})(in)$ and that $1_{s_l}$ is the indicator function for some characteristic $s_l$, $s_l \in S, 1 \leq l \leq r$; e.g., $s_1 = \{pos_{GPS} \in Germany\}$ and $s_2 = \{v_{speedometer} \leq 200km/h\}$. Next, we define the estimator $\hat{p}_{SC}^s = \prod_{l=1}^{r} s_l$. Because some characteristics of $TAS$ are not checked, $\hat{p}_{SC}^s$ systematically overestimates $p_{SC}$ and may be extended by a correction term $\varepsilon$ (which, however, needs to be defined based on expert opinion).

[1] Since we cannot obtain representative samples for all $case \notin TAS$, we make a worst-case approximation by assuming $p(m(X) = O \mid case \notin TAS) = 0$, i.e., outcomes are never correct.

**Quality impact ($p_Q$).** Besides an estimated general true-positive rate for $TAS$, quality-related knowledge for the specific input $in$ can be taken into account for the assessment of uncertainty of a single case. Information about quality-related factors (e.g., rain intensity and car velocity) is computed by $qm$, i.e., $q = qm(in)$. Next, $q$ is used by a function $qs$ to return the index $k$ of a **cluster $C_k$** of all cases $case \in TAS$ with comparable quality challenges and $o^m = 1$. Thus, we refine $p_Q$ by

$$p_{Q^{C_k}}^{o^m=1} := p(m(X) = O \mid case \in TAS, qs\big(qm(input(case))\big) = k) \tag{3}$$

Assuming $TS$ is a test dataset that is appropriately (randomly) sampled from $TAS$, i.e., each case in $TAS$ has the same probability of being part of $TS$, we can construct an estimator for $p_{Q^{C_k}}$ based on the test data belonging to cluster $c_k$. For estimating $p_{Q^{C_k}}^{o^m=1}$, the reuse of DDM training data should be avoided due to the risk of overfitting.

$$\hat{p}_{Q^{C_k}}^{o^m=1} = \frac{1}{|C_k|} \sum_{case \in C_k} \mathbf{1}_{\{case:o=1\}}(case) \tag{4}$$
$$\text{with } C_k = \{case \in TS : o^m = 1 \wedge qs\big(qm(input(case))\big) = k\}$$

The statistical uncertainty of the estimator has not been considered yet and is denoted by the expected **confidence $\alpha$** of the uncertainty wrapper. For this, we construct the Bernoulli-distributed random variable $Y := \mathbf{1}_{\{m(X)=O|X\in TAS\}}$ and compute the lower bound $\hat{p}_{Q^{C_k}}^{\alpha}$ of a single-sided confidence interval for the given confidence level (e.g. $\alpha = 0.95$). An example formula for the lower bound can be derived by the Wilson interval:

$$\hat{p}_{Q^{C_k}}^{o^m=1,\alpha} = \frac{1}{1+z_{1-\alpha}^2}\left(\hat{p}_{Q^{C_k}}^{o^m=1} + \frac{z_{1-\alpha}^2}{2} - z_{1-\alpha}\sqrt{\hat{p}_{Q^{C_k}}^{o^m=1}\left(1 - \hat{p}_{Q^{C_k}}^{o^m=1}\right) + \frac{z_{1-\alpha}{}^2}{4}}\right), \tag{5}$$

where $z_v = \Phi^{-1}(v)$ with $\Phi^{-1}$ being the quantile function of the standard normal distribution and $v \in (0,1)$. A comparison of alternative confidence intervals for the Bernoulli parameter is provided by Brown et al. [15] and the R-package 'binom' [16].

**Case-specific uncertainty ($\hat{u}^{\alpha}$).** Let $case_a$ be a case of actual application, i.e., not part of the training or the test dataset; hence, the "real" outcome $o_a$ is not available. With $in_a$ being the respective input to the DDC and $\alpha$ the requested confidence, we get a case-specific estimator for the uncertainty

$$\hat{u}^{\alpha}(case_a) \leq 1 - (\hat{p}_{SC}^{s_a} - \varepsilon)\,\hat{p}_{Q^{C_{qs(q_a)}}}^{m(x_a),\alpha} \tag{6}$$
$$\text{with } s_a = sm(in_a),\ x_a = xm(in_a),\ k_a = qm(in_a), \text{and } \varepsilon \in [0, \hat{p}_{SC}^{s_a}].$$

Note that the effect of $\alpha$ is limited to the quality-related part of uncertainty because the estimator for scope compliance cannot be statistically derived from a test dataset $TS$.

## 4 Conclusion and Outlook

The proposed uncertainty wrapper approach considers not only uncertainty caused by model fit (general misclassifications) but also case-specific uncertainty introduced by the quality of the input data and uncertainty caused by the fact that the component might

be applied in situations that were not intended when it was built and tested. In summary, this can provide a more realistic picture of uncertainty in a specific situation. The uncertainty estimate can therefore help to make better-informed decisions and initiate countermeasures if uncertainty exceeds accepted thresholds, e.g., slow down the car if the DDC is not sufficiently certain that there is no stop sign.

In the next step, we plan to instantiate the approach in a concrete case study.

**Acknowledgments.** Parts of this work is being funded by the German Ministry of Education and Research (BMBF) under grant number 01IS16043E.

## References

1. D. Solomatine and A. Ostfeld, "Data-driven modelling: some past experiences and new approaches," Journal of Hydroinformatics, vol. 10, no. 2, pp. 3-22, 2008.
2. D. Solomatine, L. See and R. Abrahart, "Data-driven modelling: concepts, approaches and experiences," Practical Hydroinformatics, pp. 17-30, 2009.
3. M. Kläs and A. M. Vollmer, "Uncertainty in Machine Learning Applications – A Practice-Driven Classification of Uncertainty," in First International Workshop on Artificial Intelligence Safety Engineering (WAISE 2018), 2018.
4. J. S. Armstrong, "Chapter: The Forecasting Dictionary," in Principles of forecasting: a handbook for researchers and practitioners, Springer Science & Business Media, 2001.
5. M. Kläs, "Towards Identifying and Managing Sources of Uncertainty in AI and Machine Learning Models - An Overview," arXiv preprint arXiv:1811.11669, 2018.
6. S. Lee and W. Chen, "A comparative study of uncertainty propagation methods for black-box-type problems," Structural and Multidisciplinary Optimization, vol. 37, p. 239, 2009.
7. S. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, no. 3, pp. 660-674, 1991.
8. A. Khosravi, S. Nahavandi, D. Creighton and A. Atiya, "Comprehensive review of neural network-based prediction intervals and new advances," IEEE Transactions on neural networks, vol. 22, no. 9, pp. 1341-1356, 2011.
9. Y. Gal, "Uncertainty in deep learning," University of Cambridge, 2016.
10. R. McAllister, Y. Gal, A. Kendall, M. Van Der Wilk, A. Shah, R. Cipolla and A. Weller, "Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning," in International Joint Conferences on Artificial Intelligence, 2017.
11. M. Kläs, A. Trendowicz, A. Wickenkamp, J. Münch, N. Kikuchi and Y. Ishigai, "The use of simulation techniques for hybrid software cost estimation and risk analysis," Advances in computers, vol. 74, pp. 115-174, 2008.
12. L. Angelis and I. Stamelos, "A simulation tool for efficient analogy based cost estimation," Empirical software engineering, vol. 5, no. 1, pp. 35-68, 2000.
13. D. Shrestha and D. Solomatine, "Machine learning approaches for estimation of prediction interval for the model output," Neural Networks, vol. 19, no. 2, pp. 225-235, 2006.
14. D. Solomatine and D. Shrestha, "A novel method to estimate model uncertainty using machine learning techniques," Water Resoures Research, vol. 45, no. 12, p. W00B11, 2009.
15. L. Brown , T. Cai, A. DasGupta, "Interval Estimation for a Binomial Proportion," Statistical Science. vol. 16, no. 2, pp. 101-133, 2001.
16. S. Dorai-Raj, "Cran R packages: 'binom'," February 2015. [Online]. Available: https://cran.r-project.org/web/packages/binom/binom.pdf. [Accessed May 2019].