Explainability through uncertainty: Trustworthy decision-making with neural networks

Arthur Thuy[a,b,*], Dries F. Benoit[a,b]

[a] *Ghent University, Research Group Data Analytics, Faculty of Economics and Business Administration, Tweekerkenstraat 2, 9000 Ghent, Belgium*

[b] *CVAMO Core Lab, Flanders Make, Gaston Geenslaan 8, 3001 Leuven, Belgium*

Abstract

Uncertainty is a key feature of any machine learning model and is particularly important in neural networks, which tend to be overconfident. This overconfidence is worrying under distribution shifts, where the model performance silently degrades as the data distribution diverges from the training data distribution. Uncertainty estimation offers a solution to overconfident models, communicating when the output should (not) be trusted. Although methods for uncertainty estimation have been developed, they have not been explicitly linked to the field of explainable artificial intelligence (XAI). Furthermore, literature in operations research ignores the actionability component of uncertainty estimation and does not consider distribution shifts. This work proposes a general uncertainty framework, with contributions being threefold: (i) uncertainty estimation in ML models is positioned as an XAI technique, giving local and model-specific explanations; (ii) classification with rejection is used to reduce misclassifications by bringing a human expert in the loop for uncertain observations; (iii) the framework is applied to a case study on neural networks in educational data mining subject to distribution shifts. Uncertainty as XAI improves the model's trustworthi-

[*] Corresponding author

Email addresses: arthur.thuy@ugent.be (Arthur Thuy), dries.benoit@ugent.be (Dries F. Benoit)

ness in downstream decision-making tasks, giving rise to more actionable and robust machine learning systems in operations research.

Keywords: Decision support systems, Explainable artificial intelligence, Monte Carlo Dropout, Deep Ensembles, Distribution shift

---

1. Introduction

A representation of uncertainty is desirable and is a key feature of any machine learning (ML) model. Uncertainty is particularly important in neural networks (NNs), which tend to be overconfident in their predictions (Guo et al., 2017). That is, a NN classifier often predicts an incorrect label, despite giving a high predicted probability.

This flaw is especially troubling in situations of distribution shift, where the data distribution during deployment diverges from the training data distribution (Murphy, 2022). Although the model performs well when first deployed, its performance degrades over time as the distribution shift increases without warning the decision maker. Distribution shifts happen all the time, either suddenly, gradually, or seasonally (Huyen, 2022). For example, a demand prediction model is affected by a sudden change in the pricing policy of a competitor or when a new competitor enters the market.

The field of uncertainty estimation provides a solution to overconfident models by capturing the uncertainty in both the data and the model. As such, it communicates when a model's output should (not) be trusted (Ovadia et al., 2019). Building trust is also the cornerstone of the field of explainable artificial intelligence (XAI), which aims to explain the output of black-box models. XAI techniques are commonly used in operations research (OR) to facilitate the human-computer interaction and thereby support decision-making systems (Cabitza et al., 2023).

The related work on uncertainty estimation and XAI has three shortcomings: (i) uncertainty estimation is not explicitly formulated as an XAI technique following the local/global and model-specific/agnostic specification

and there is no theoretical motivation on how uncertainty contributes to explainability; (ii) the available work in OR merely monitors the NN uncertainty estimates without acting upon it; (iii) there is a lack of OR applications that examine the influence of distribution shifts on NN uncertainty, as literature only employs benchmark datasets like MNIST.

A general uncertainty framework is proposed, with contributions being threefold:

1. The framework first positions uncertainty estimation in ML models as an *XAI technique*, giving local and model-specific explanations. To support this, theoretical properties are discussed, arguing that uncertainty estimation fosters higher levels of *trust*, *actionability*, and *robustness*.
2. The framework then uses *classification with rejection* (Mena et al., 2021) to reduce misclassifications by bringing a human expert in the loop for uncertain observations.
3. The framework is applied to a case study on *neural networks* in educational data mining, with distribution shifts occurring naturally when deploying the model to production.

The remainder of the paper is organized as follows. Section 2 gives an overview of related work and identifies shortcomings. Section 3 presents the general uncertainty framework and positions uncertainty estimation as an XAI technique. Section 4 discusses how uncertainty is quantified specifically in NN classifiers. In section 5, the case study in educational data mining with NN uncertainty is presented; section 6 gives the results. Finally, section 7 provides a discussion and section 8 gives a conclusion.

2. Related Work

This section discusses related work on XAI, uncertainty estimation, and NNs in the field of OR. Furthermore, extant literature on uncertainty estimation as XAI is discussed. Thereby, three main shortcomings in related work are identified.

2.1. Explainable artificial intelligence in operations research

ML models are widely used in OR to solve complex problems (Choi et al., 2018). However, extant literature often focuses on predictive performance which comes at the expense of model explainability. This lack of explainability leads to decision makers' distrust and unwillingness to adopt analytics in decision support systems (Shin, 2021).

The field of XAI refers to techniques that try to explain how a black-box ML model produces its outcomes. Although still limited, XAI techniques are increasingly adopted in OR applications, e.g., in credit risk (Bastos & Matos, 2022; Sachan et al., 2020), marketing risk (De Caigny et al., 2018; Van Nguyen et al., 2020), supply chain management (Garvey et al., 2015), healthcare (Piri et al., 2017), and jurisprudence (Delen et al., 2021). As such, XAI bridges the gap to organizational decision-makers by providing understanding into a model's predictions and generating actionable insights.

XAI techniques can be organized based on two main criteria (Adadi & Berrada, 2018). It can be global, i.e., characterize the whole dataset (e.g., partial dependence plot), or local, i.e., explain individual predictions (e.g., counterfactual explanations). It can be model-specific, i.e., capable of explaining only a restricted class of models (e.g., random forest variable importance), or model-agnostic, i.e., applicable to any model (e.g., SHAP).

2.2. Uncertainty and neural networks in operations research

Neural networks are rapidly emerging in operations research (OR), with applications such as credit scoring, demand prediction, and outlier detection (Kraus et al., 2020; Gunnarsson et al., 2021; Verboven et al., 2021; Van Belle et al., 2021). Kraus et al. (2020) point to three key challenges that limit the relevance of deep learning in OR: (i) extensive hyperparameter tuning is required, (ii) lack of uncertainty estimation, and (iii) lack of accountability and explainability.

Uncertainty estimation for NNs has been investigated in different domains of OR: predictive maintenance (Kraus & Feuerriegel, 2019), recommender systems (Nahta et al., 2021), finance (Ghahtarani, 2021), stress-level prediction

(Oh et al., 2021), transportation (Zhang & Mahadevan, 2020; Feng et al., 2022), predictive process monitoring (Weytjens & De Weerdt, 2022), and educational data mining (Yu et al., 2021). In the available work, however, uncertainty estimates are merely monitored as an additional metric, not used in combination with a human expert such as in *classification with rejection* (i.e., shortcoming 1). Ignoring the actionability of this human-machine combination leaves a large part of the added value on the table.

Moreover, there is a lack of literature on the impact of distribution shifts on NN uncertainty estimates with applications in OR (i.e., shortcoming 2). That is, uncertainty estimates are always evaluated on benchmark datasets, e.g., MNIST and not-MNIST, or using artificial distortions, e.g., Gaussian blur (Ovadia et al., 2019).

2.3. Uncertainty as explainable artificial intelligence

Bai et al. (2021) are the first to list uncertainty estimation as a third distinct category in XAI, next to attribution-based (e.g., SHAP) and non-attribution-based (e.g., counterfactual explanations) methods (Figure 1). However, Bai et al. (2021) do not specify uncertainty estimation in terms of the two main XAI criteria and do not provide a theoretical motivation (i.e., shortcoming 3).

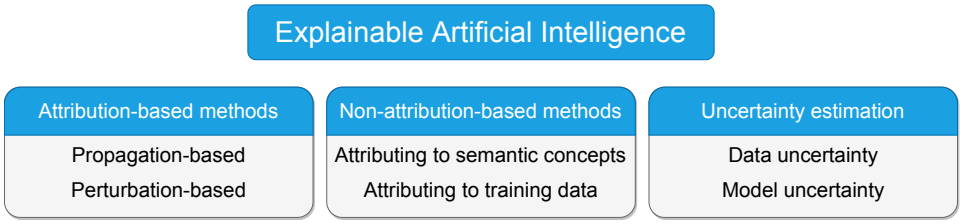

Figure 1: **Overview explainable artificial intelligence.** Uncertainty estimation is a third general type of XAI technique. Figure adapted from Bai et al. (2021).

This work addresses the three shortcomings by proposing a general uncertainty framework, positioning *uncertainty estimation in ML models as XAI* and using *classification with rejection*. Furthermore, a case study on *NNs* with distribution shifts demonstrates the value of the framework for OR applications.

3. Methodology

Figure 2 outlines the proposed general uncertainty framework. The framework consists of two stages: (i) uncertainty estimation as XAI and (ii) classification with rejection. The goal of uncertainty estimation is to quantify the data and model uncertainty in predictions made by an ML model. The classification with rejection system then uses the estimates to assist in deciding which predictions should be rejected or retained, based on three key metrics.

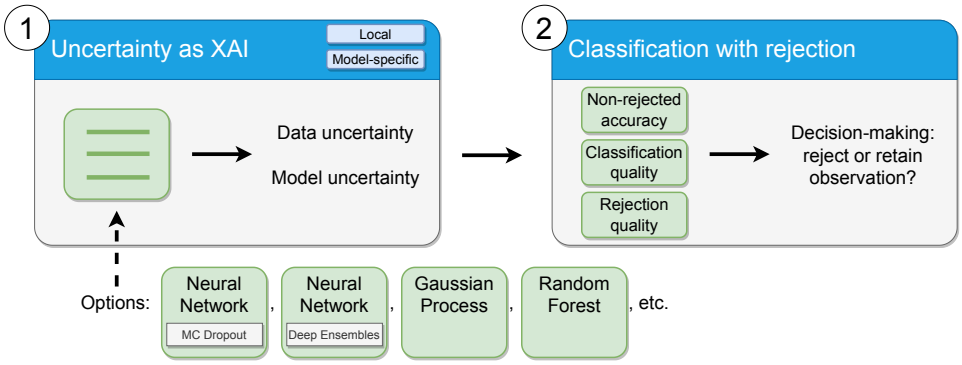

Figure 2: **General uncertainty framework.** The framework consists of two stages: (i) uncertainty estimation as XAI and (ii) classification with rejection. It can be applied to multiple ML models, each having one or more specific uncertainty techniques.

3.1. Uncertainty as explainable artificial intelligence

Uncertainty as XAI is available for multiple ML models, each having distinct techniques. That is, the case study quantifies data and model uncertainty in NNs, but it can also be computed in e.g., Gaussian Processes (Hüllermeier & Waegeman, 2021) or Random Forests (Shaker & Hüllermeier, 2020) using other existing techniques. Furthermore, one can even use different uncertainty estimation techniques for some ML models, e.g., Monte Carlo Dropout and Deep Ensembles for NNs (Gal & Ghahramani, 2016; Lakshminarayanan et al., 2017).

3.1.1. Data and model uncertainty

Each prediction has two uncertainty values, as uncertainty can arise from two fundamentally different sources: data uncertainty and model uncer-

tainty (Der Kiureghian & Ditlevsen, 2009). *Data uncertainty*, also known as aleatoric uncertainty, refers to the notion of randomness and is related to the data-measurement process. This uncertainty is irreducible even if more data is collected. *Model uncertainty*, also known as epistemic uncertainty, accounts for uncertainty in the model parameters, i.e., uncertainty about which model generated the collected data. In contrast to data uncertainty, collecting more data can reduce model uncertainty. Both types of uncertainty can then be summed to compute the *total uncertainty* in a prediction.

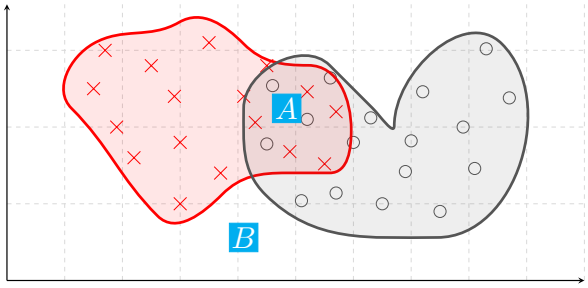

Figure 3: **Two types of uncertainty.** Observation A has high data uncertainty; B has high model uncertainty. Figure adapted from Hüllermeier & Waegeman (2021).

Consider a binary classification task with two input features (Figure 3), where the crosses represent positive training examples and the circles represent negative training examples. At test time, predictions are made for both observations A and B. The model uncertainty is high in sparsely populated regions with few training examples. Therefore, observation A has high model uncertainty and could be classified as either positive or negative. In contrast, observation B lies in a region where the two class distributions are overlapping, i.e., the data uncertainty is high. Although collecting more training data around observation B will reduce the model uncertainty, more training data around observation A will not reduce the data uncertainty.

3.1.2. Theoretical properties

Doshi-Velez & Kim (2017) devise six desirable properties for XAI techniques: trust, actionability, fairness, privacy, robustness, and causality. We apply

uncertainty estimation to this list and argue that it satisfies three properties:

- Trust: decision makers should feel comfortable relinquishing control to the ML model. As model uncertainty enables saying "I do not know," a human expert can step in. This awareness gives decision makers more confidence to rely on the model's predictions in other situations when it says "I do know."
- Actionability: ML models should provide information assisting users to accomplish a task. Uncertainty estimates are key in classification with rejection, where uncertain observations are passed on to a human expert.
- Robustness: ML models should reach certain levels of performance in the face of input variation. Under increased distribution shift, uncertainty estimates grow accordingly, enabling an improvement in accuracy by rejecting the most uncertain observations.

To demonstrate the validity of the theoretical properties, they are evaluated in light of the case study results (see section 7). Uncertainty is complementary to other XAI techniques, which can be used to satisfy the remaining three properties.

3.1.3. Specification

Uncertainty estimation is a local and model-specific XAI method. It is *local* because each observation receives an uncertainty estimate, both for data and model uncertainty. Furthermore, it is *model-specific* because techniques for decomposition into data and model uncertainty are different across ML models, although they exist for multiple ML models.

3.2. Classification with rejection

For predictions with high uncertainty, the observations can be passed on to a human expert for a label. The goal of a *classification with rejection* system (Mena et al., 2021; Barandas et al., 2022) is to help decide when to stop rejecting the most uncertain observations. The system takes as

input the per-observation uncertainty values and outputs three metrics to assist the decision maker in finding the optimal rejection threshold for the task at hand. It is useful for applications where making an error can be more costly than asking a human expert for help. For example, in fraud detection, an employee can verify a transaction manually if the prediction is uncertain.

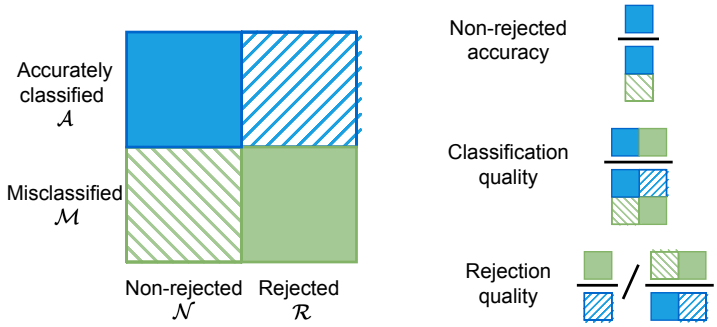

Figure 4: **Performance measures for classification with rejection.** Three performance measures are proposed by Condessa et al. (2017) to find the optimal rejection point. Figure adapted from Mena et al. (2021).

Observations are classified along two criteria: (i) accurately classified $\mathcal{A}$ and misclassified $\mathcal{M}$; (ii) rejected $\mathcal{R}$ and non-rejected $\mathcal{N}$. Condessa et al. (2017) propose three rejection metrics (Figure 4; higher is better):

- Non-rejected accuracy (NRA): ability to classify non-rejected samples accurately.

$$NRA = \frac{|\mathcal{A} \cap \mathcal{N}|}{|\mathcal{N}|} \tag{1}$$

- Classification quality (CQ): ability to retain correctly classified samples and to reject misclassified samples, i.e., correct decision-making.

$$CQ = \frac{|\mathcal{A} \cap \mathcal{N}| + |\mathcal{M} \cap \mathcal{R}|}{|\mathcal{N}| + |\mathcal{R}|} \tag{2}$$

- Rejection quality (RQ): ability to concentrate misclassified samples in

the set of rejected samples.

$$RQ = \frac{|\mathcal{M} \cap \mathcal{R}|}{|\mathcal{A} \cap \mathcal{R}|} / \frac{|\mathcal{M}|}{|\mathcal{A}|} \tag{3}$$

The NRA and CQ are bounded in the interval $[0, 1]$, unlike the RQ which has a minimum value of zero and an unbounded maximum. The three metrics are evaluated as a function of the percentage of rejections varying from 0% to 100% (Yong & Brintrup, 2022). If observations have identical uncertainty values (e.g., exactly 0.0 or 1.0), observations are rejected randomly until the desired percentage is achieved.

3.3. Workflow with a human-in-the-loop

The suggested way of working for the human expert is as follows. The accuracy score is first evaluated on the entire test set, without rejecting any observations. If the accuracy is not sufficiently high, the rejection process is started and the metrics NRA, CQ, and RQ are evaluated. The most uncertain observations are rejected until the labeling budget for the expert annotator is exhausted, or until the NRA is sufficiently high. Although the NRA is the most important metric, the CQ and RQ provide more information on the internals of the rejection system. For example, a decreasing CQ indicates that more and more correct observations are rejected. At this point, the expert might decide to stop rejecting because the NRA will likely stagnate, which does not justify spending the labeling budget on.

It is important to note that the rejection level depends on the labeling budget available for the expert annotator and the accuracy requirement, associated with the misclassification cost, for the task. As such, there is no universally optimal point of rejection.

4. Uncertainty in Neural Networks

In the case study, NNs are used for the "Uncertainty as XAI" building block of the general uncertainty framework. This section discusses how uncertainty can be represented and measured in NNs.

4.1. Data and model uncertainty

Data uncertainty. In a NN classifier, the output layer contains a softmax or sigmoid function, forming a categorical distribution over the class labels $p(\mathbf{Y} \mid \mathbf{X},\ \boldsymbol{\theta})$. This distribution enables the NN to represent *data uncertainty*.

Modern NNs are usually trained using a maximum likelihood objective. That is, they find a single setting of parameters $\boldsymbol{\theta}^*$ to maximize the probability of the data given the parameters, $\arg\max_{\boldsymbol{\theta}} p(\mathbf{X},\ \mathbf{Y} \mid \boldsymbol{\theta})$. For each test input $\mathbf{x}^*$, there is only one prediction because the NN generates an identical output for each run. As a result, model uncertainty cannot be captured.

Model uncertainty. NNs are large flexible models capable of representing many functions, corresponding to different parameter settings. Each function fits the training data well, yet generalizes in different ways, a phenomenon known as *underspecification* (Wilson, 2020). Considering all of these different NNs together allows capturing *model uncertainty*. In a probabilistic sense, uncertainty in an unseen input point $\mathbf{x}^*$ is represented by the posterior predictive distribution $p(\mathbf{y}^* \mid \mathbf{x}^*,\ \mathbf{X},\ \mathbf{Y})$.

Model uncertainty can be captured in NNs using two approaches: (i) Bayesian NNs and (ii) ensembles. A Bayesian NN aims to estimate a full distribution for $p(\boldsymbol{\theta} \mid \mathbf{X},\ \mathbf{Y})$, unlike maximum likelihood. However, this distribution is intractable and is typically approximated using sampling techniques. The ensembling approach obtains multiple good maximum likelihood settings $\boldsymbol{\theta}^*$.

Both approaches aggregate predictions over a collection of NNs. The following subsections discuss the most popular technique for either approach, (i) Monte Carlo Dropout and (ii) Deep Ensembles. Figure 5 provides a visual overview.

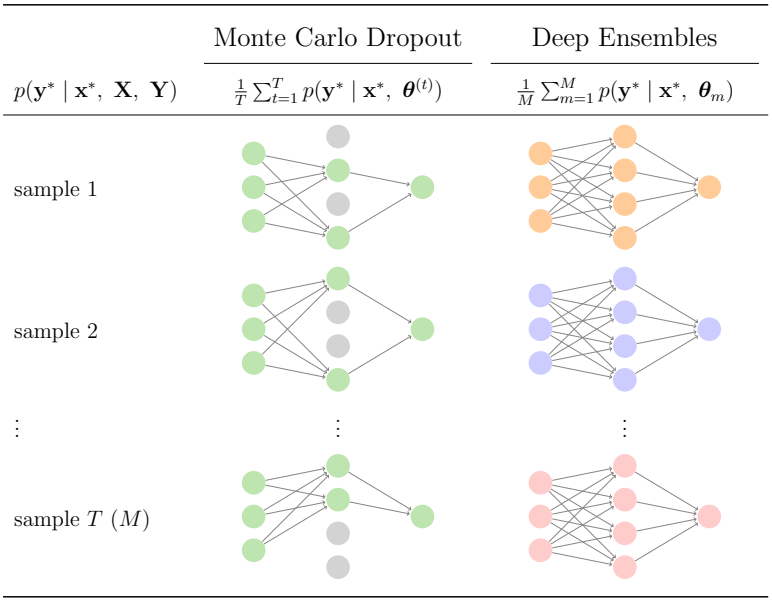

Figure 5: **Overview of uncertainty estimation methods.** Forward passes are generated differently depending on the method. In MC dropout, different units are dropped out from a NN; in Deep Ensembles, multiple independent NNs are used, with different parameter initializations and noise in the SGD training process.

4.2. Monte Carlo Dropout

In Monte Carlo (MC) Dropout (Gal & Ghahramani, 2016), dropout is not only applied at training time but also at test time. Multiple forward passes are performed, each time randomly dropping units and getting another thinned dropout variant of the NN. The resulting T predictions $\{\widehat{\mathbf{y}}_1^*(\mathbf{x}^*),\ \ldots,\ \widehat{\mathbf{y}}_T^*(\mathbf{x}^*)\}$ are aggregated, forming an approximation to the true posterior predictive distribution:

$$p(\mathbf{y}^* \mid \mathbf{x}^*,\ \mathbf{X},\ \mathbf{Y}) = \int p(\mathbf{y}^* \mid \mathbf{x}^*,\ \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathbf{X},\ \mathbf{Y})\, d\boldsymbol{\theta} \tag{4}$$

$$\approx \frac{1}{T}\sum_{t=1}^{T} p(\mathbf{y}^* \mid \mathbf{x}^*,\ \boldsymbol{\theta}^{(t)}). \tag{5}$$

The posterior predictive distribution is obtained through Bayesian model averaging. That is, it averages over an infinite collection of parameter settings, weighted by their posterior probabilities.

4.3. Deep Ensembles

Deep Ensembles (Lakshminarayanan et al., 2017) uses an ensemble of M maximum likelihood NNs, with every NN trained on the same dataset and the same input features. The diversity arises through different parameter initializations and noise in the stochastic gradient descent (SGD) training process, inducing different solutions due to the non-convex loss. At test time, each of the M NNs performs one forward pass. The resulting M predictions $\{\widehat{\mathbf{y}}_1^*(\mathbf{x}^*),\ \ldots,\ \widehat{\mathbf{y}}_M^*(\mathbf{x}^*)\}$ are averaged, forming a mixture distribution:

$$p(\mathbf{y}^* \mid \mathbf{x}^*) \approx \frac{1}{M} \sum_{m=1}^{M} p(\mathbf{y}^* \mid \mathbf{x}^*,\ \boldsymbol{\theta}_m). \tag{6}$$

In ensembling, NNs are weighted equally over a finite collection of functions. As such, it is a fundamentally different mindset than Bayesian model averaging.

4.4. Uncertainty decomposition

The posterior predictive distribution holds information about the total uncertainty in a prediction, decomposable in data and model uncertainty using classical information-theoretic measures. However, calculations require the expectation over the posterior distribution, which is intractable. Nonetheless, an approximation can be obtained using samples from the approximate posterior predictive distribution:

$$u_{total}(\mathbf{x}^*) \approx H\left[\frac{1}{T}\sum_{t=1}^{T} p(\mathbf{y}^* \mid \mathbf{x}^*,\ \boldsymbol{\theta}^{(t)})\right] \tag{7}$$

$$u_{data}(\mathbf{x}^*) \approx \frac{1}{T}\sum_{t=1}^{T} H\left[p(\mathbf{y}^* \mid \mathbf{x}^*,\ \boldsymbol{\theta}^{(t)})\right] \tag{8}$$

$$u_{model}(\mathbf{x}^*) = u_{total}(\mathbf{x}^*) - u_{data}(\mathbf{x}^*). \tag{9}$$

First, total uncertainty and data uncertainty are calculated; then model uncertainty is obtained as the difference (Depeweg et al., 2018). Total uncertainty is computed by averaging over the different samples and calculating the entropy H. Data uncertainty is computed by calculating the entropy in each sample and averaging the entropies. This boils down to fixing a set of weights $\boldsymbol{\theta}^{(t)}$, i.e., considering a distribution $p(\mathbf{y}^* \mid \mathbf{x}^*,\ \boldsymbol{\theta}^{(t)})$, essentially removing the model uncertainty. Model uncertainty is high if the distribution $p(\mathbf{y}^* \mid \mathbf{x}^*,\ \boldsymbol{\theta}^{(t)})$ varies greatly for different weights $\boldsymbol{\theta}^{(t)}$. Intuitively, data uncertainty measures uncertainty in the softmax classification on individual samples; model uncertainty measures how much the samples deviate (Hüllermeier & Waegeman, 2021; Barandas et al., 2022).

Table 1: **Examples of uncertainty decomposition.** The middle and bottom row have equal total uncertainty but have wildly different samples. Decomposition in data and model uncertainty reveals the different characteristics.

Samples $p(\mathbf{y}^* \mid \mathbf{x}^*,\ \boldsymbol{\theta}^{(t)})$	$p(\mathbf{y}^* \mid \mathbf{x}^*)$	$u_{total}(\mathbf{x}^*)$	$u_{data}(\mathbf{x}^*)$	$u_{model}(\mathbf{x}^*)$
{(1. , 0.), (1. , 0.), (1. , 0.), (1. , 0.)}	(1. , 0.)	0.	0.	0.
{(0.5, 0.5), (0.5, 0.5), (0.5, 0.5), (0.5, 0.5)}	(0.5, 0.5)	1.	1.	0.
{(1. , 0.), (0. , 1.), (1. , 0.), (0. , 1.)}	(0.5, 0.5)	1.	0.	1.

Table 1 contains three examples in the context of binary classification with $T = 4$ samples. The middle and bottom rows both have a total uncertainty of 1.0 although the samples are wildly different. Therefore, the total uncertainty alone is not sufficient to characterize the NN's predictions; decomposition into data and model uncertainty is necessary.

5. Case Study: Student Performance Prediction

5.1. Problem setting

Student performance prediction is extensively discussed in OR literature (Delen et al., 2020; Coussement et al., 2020; Olaya et al., 2020; Deeva et al., 2022; Phan et al., 2023). Common performance metrics include student dropout, course certification, final course grade, pass/fail, etc. Developing predictive models for student performance forms the basis for an educational

early-warning system, where at-risk students are identified on time and assisted with personalized support by course advisors. Therefore, in order to deliver support, a predictive model should provide predictions being both *accurate* and *actionable*.

Whitehill et al. (2017) and Gardner & Brooks (2018) argue that most prior research has poor actionability due to same course–same year evaluation. This training paradigm creates a practical problem because the target labels required by supervised learning algorithms only become available after the final exam, when any support for students is too late. Alternatives that resolve this issue are training on a previous edition of the course, or training on a different course altogether if there is no previous edition available.

The case study applies the uncertainty framework to NNs and investigates uncertainty estimation as an XAI technique in a predictive setup subject to distribution shifts. The results are compared to a standard NN only capable of capturing data uncertainty, but no model uncertainty. The experiment answers the call of Gašević et al. (2016) for research on changing course conditions in student performance prediction, advocating that learning analytics should account for the fluid nature of technology use within a course.

5.2. Data

The dataset ([dataset] MITx & HarvardX, 2014) consists of student-course records of HarvardX and MITx massive open online courses (MOOCs) hosted on the edX platform about a wide range of topics, over two semesters (fall 2012 and spring 2013). The binary target labels denote whether a student scored a grade high enough to earn a certificate; features include processed clickstream activities and student demographics.

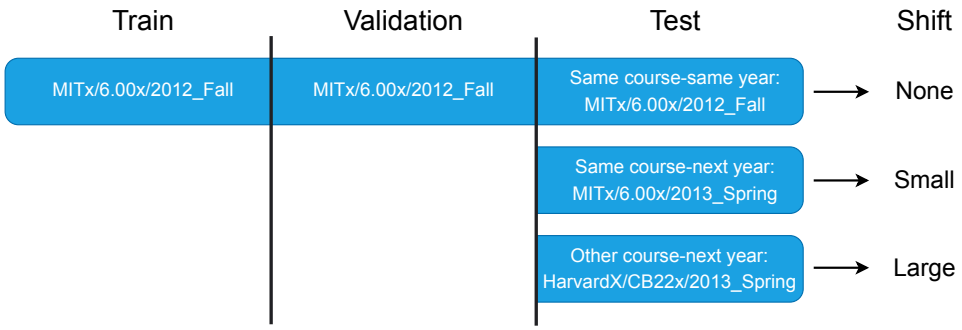

Figure 6: **Experimental setup.** A NN is trained on the course "MITx/6.00x/2012_Fall". Next, predictions are made for three test sets: (i) "MITx/6.00x/2012_Fall" (same course–same year), (ii) "MITx/6.00x/2013_Spring" (same course–next year), (iii) "HarvardX/CB22x/2013_Spring" (other course–next year).

5.3. Experimental setup

The experimental setup is detailed in Figure 6. First, the NN is trained on the course "MITx/6.00x/2012_Fall", denoting the MITx course 6.00x "Introduction to Computer Science and Programming" of fall 2012. This course is selected because it has the largest number of observations and is offered in both semesters. Next, predictions are made on three test sets: (i) same course–same year: "MITx/6.00x/2012_Fall", (ii) same course–next year: "MITx/6.00x/2013_Spring", and (iii) other course–next year: "HarvardX/CB22x/2013_Spring". Course CB22x is titled "The Ancient Greek Hero" and is selected as an extreme case because it is a non-STEM course offered by a different university. It is important to note that the input features gathered for HarvardX and MITx courses are identical because they are both hosted on the edX platform.

Predicting on the three test sets represents three different distribution shifts. This ranges from (i) no shift (same course–same year), to (ii) small shift (same course–next year), to (iii) large shift (other course–next year). The situation of no shift serves as a baseline because it is often used in literature, despite being practically infeasible.

The data include students who accessed at least half of the chapters in the course material, with the training set having a class distribution of 56/44

and 2000 observations. The three test sets have a class distribution of (i) 56/44, (ii) 46/54, and (iii) 72/28, respectively.

All three methods (standard, MC Dropout, Deep Ensembles) use the same NN configuration as the building block. The NN is a multi-layer perceptron with 2 hidden layers, each containing 64 hidden units, a ReLU activation function, a dropout rate of 0.4 and 0.5, and Glorot uniform weight initialization. The NNs are trained with the Adam optimizer and a binary cross-entropy loss function for 50 epochs with a batch size of 32 and a learning rate of 5×10^{-4}, using early stopping. For MC Dropout, the NN predicts 128 samples per input observation. For Deep Ensembles, 10 NNs are trained, resulting in 10 samples per input observation. Results are averaged over 10 runs with random data splits.

6. Results

6.1. Total uncertainty

Figures 7, 8, and 9 show the histograms of the uncertainty distributions for all three methods on all three shifts (rows); the total uncertainty is decomposed into data and model uncertainty (columns). The vertical dashed line denotes the mean of the uncertainty values, which can be used to quickly compare centrality across distributions.

With increased distribution shift, data uncertainty consistently decreases for the standard NN, i.e., more mass is located at low uncertainty values and the mean value decreases. For MC Dropout and Deep Ensembles, data uncertainty remains equal when moving to the small shift before decreasing substantially on the large shift. In contrast to data uncertainty, model uncertainty grows rapidly for MC Dropout and Deep Ensembles. The standard NN cannot capture model uncertainty (i.e., value zero for all observations) and only relies on the decreasing data uncertainty to calculate total uncertainty.

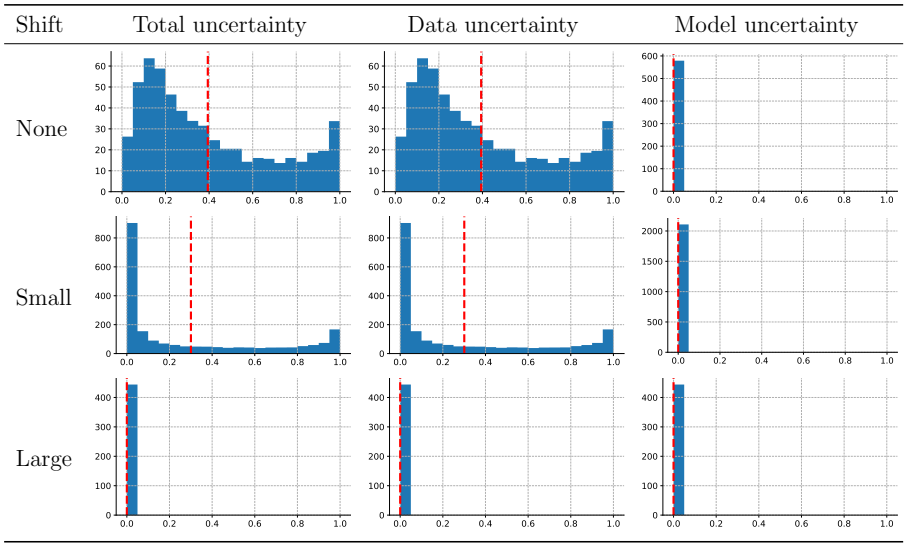

Figure 7: **Standard: uncertainty distributions.** The histogram displays absolute frequency and the dashed line denotes the mean value. With increased distribution shift, data uncertainty decreases so total uncertainty decreases as well because the standard NN does not capture model uncertainty.

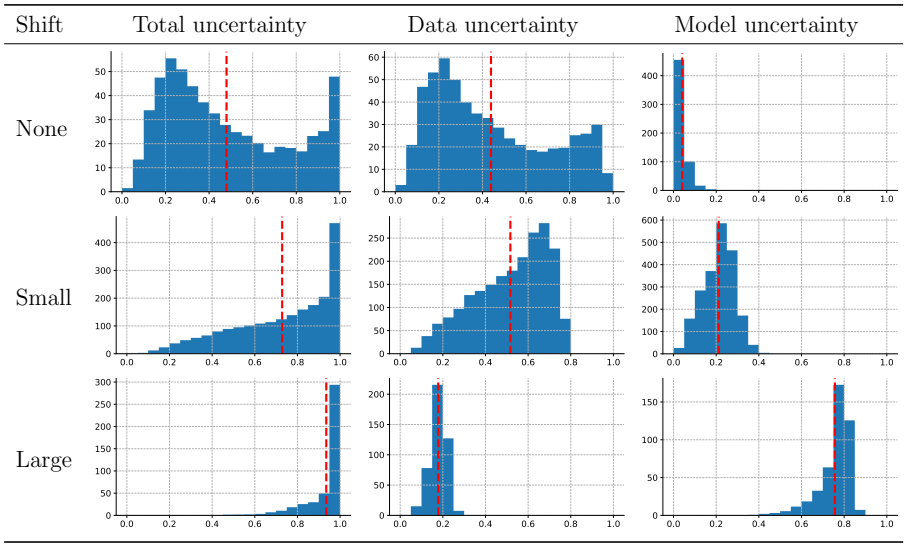

Figure 8: **MC Dropout: uncertainty distributions.** The histogram displays absolute frequency and the dashed line denotes the mean value. With increased distribution shift, data uncertainty stagnates or decreases while model uncertainty increases consistently. As a result, MC Dropout has increased total uncertainty.

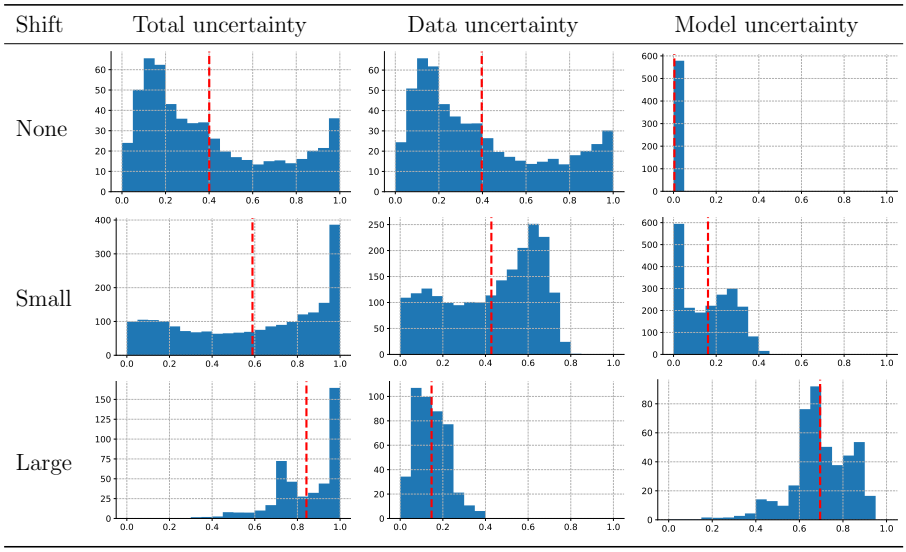

Figure 9: **Deep Ensembles: uncertainty distributions.** The histogram displays absolute frequency and the dashed line denotes the mean value. With increased distribution shift, data uncertainty stagnates or decreases while model uncertainty increases consistently. As a result, Deep Ensembles has increased total uncertainty.

In summary, for increased distribution shift, the standard NN has decreased total uncertainty, whereas MC Dropout and Deep Ensembles have rapidly increased total uncertainty. In other words, the standard NN becomes more confident as the inputs stray away from the training data distribution, which is undesirable behavior. This is in contrast to MC Dropout and Deep Ensembles, which indicate that the NN "knows what it does not know."

Table 2: **Accuracy (%).** Accuracy degrades with increased distribution shift for all NNs. For a small and large shift, MC Dropout and Deep Ensembles outperform the standard NN. Mean ± standard error are reported.

Shift	Standard	MC Dropout	Deep Ensembles
None	88.17 ± 0.10	87.81 ± 0.25	88.53 ± 0.11
Small	63.85 ± 0.92	84.43 ± 1.04	84.93 ± 0.52
Large	27.93 ± 0.00	45.05 ± 4.30	31.01 ± 1.57

6.2. Accuracy

Table 2 shows the accuracy for all three methods (columns) on all three shifts (rows). It is important to note that MC Dropout and Deep Ensembles average over the different samples to get the final probability vector, capturing model uncertainty.

On increasingly shifted data, the accuracy degrades for all three methods, as expected. For no shift, MC Dropout has a slightly lower accuracy than the standard NN while Deep Ensembles performs slightly better. For a large shift, all NNs perform poorly because they tend to naively predict the minority class, which illustrates how difficult the task is. The results of MC Dropout also have a larger standard error in this situation, indicating that the model results alternate between predicting the minority class and making more sensible predictions. In the situation of a small shift, information contained in different samples (i.e., model uncertainty) has a big impact on accuracy; MC Dropout and Deep Ensembles improve substantially over the standard NN. That is, the standard NN has an accuracy of 63.85%, MC Dropout has 84.43%, and Deep Ensembles has 84.93%.

It is worth noting that although all methods have poor accuracy under a large shift, MC Dropout and Deep Ensembles raise a warning through increased uncertainty whereas the standard NN is confidently wrong. The uncertainty values are then used to reject the most uncertain observation, i.e., classification with rejection.

6.3. Non-rejected accuracy

Figure 10 (left column) displays the non-rejected accuracy for all three methods on all three shifts (rows), based on the total uncertainty. Note that the curve at rejection 0.0% corresponds to the method's accuracy without rejection in Table 2.

In the situation of no shift (top row), all three methods achieve 100% accuracy. The same holds for a small shift (middle row), despite that the standard NN started at a substantially lower initial accuracy. For a large shift (bottom row), initial accuracies are all poor but only MC Dropout and

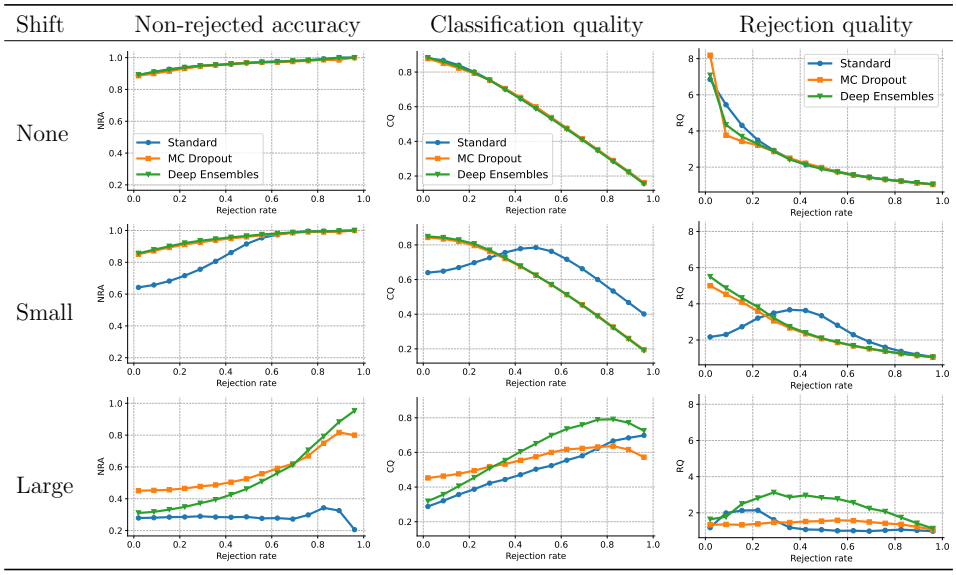

Figure 10: **Classification with rejection.** NRA, CQ, and RQ are displayed for increased distribution shifts, for all three models. Under a large distribution shift, the standard NN's uncertainty estimates are uninformative and rejections are random. This is evidenced by the stagnating NRA, ever-increasing CQ and low RQ.

Deep Ensembles manage to increase the accuracy by rejecting the most uncertain observations. Deep Ensembles performs better with 95% accuracy at rejection rate 0.95, whereas MC Dropout only obtains 80% accuracy. The standard NN, in contrast, has a large amount of observations with uncertainty zero. Since these uncertainty values are identical, observations are rejected randomly, causing the non-rejected accuracy to stagnate at the initial accuracy.

Only MC Dropout and Deep Ensembles have informative uncertainty values so that appropriate observations are rejected, effectively increasing the NRA. The standard NN, on the other hand, fails to increase the NRA under large distribution shifts.

6.4. Classification quality

Figure 10 (middle column) shows the classification quality for all three methods on all three shifts (rows), based on the total uncertainty. Classification

quality measures the correct decision-making of the classifier-rejector; accurately classified samples should be retained and misclassified samples should be rejected. In other words, the curve shows where the maximum number of correct decisions is made.

In the situation of no shift (top row), all three curves decrease gradually, indicating that the majority of rejected observations are correctly classified. As such, it is optimal to not reject any observations. When a distribution shift is present (middle and bottom row), MC Dropout and Deep Ensembles obtain the point of optimal decision-making at smaller rejection rates than the standard NN. For the small shift (middle row), MC Dropout and Deep Ensembles obtain the maximum CQ at rejection rate 0%; the standard NN needs 50% rejections.

For the large shift (bottom row), the CQ curve of the standard NN keeps on increasing, i.e., it would be best to reject all observations. In contrast, MC Dropout and Deep Ensembles obtain the maximum value at rejection rate 80%. At this rate, Deep Ensembles outperforms MC Dropout with a CQ of 0.80, as compared to MC Dropout's CQ of 0.63. These findings indicate that the uncertainty estimates of the standard NN are least effective at deciding whether observations should be retained or rejected, and that Deep Ensembles is more effective than MC Dropout.

6.5. Rejection quality

Figure 10 (right column) displays the rejection quality for all three methods on all three shifts (rows), based on the total uncertainty. Rejection quality measures the ability to reject misclassified samples. That is, it compares the proportion of misclassified to accurately classified samples on the set of rejected samples with that proportion on the entire data set.

In the situation of no shift (top row), all three curves decrease rapidly, indicating that the proportion of misclassified observations in the rejected set decreases as more observations are rejected, which is undesirable. For the small shift (middle row), MC Dropout and Deep Ensembles obtain the

highest RQ at rejection rate 0%. In contrast, the standard NN requires 40% rejections to do the same.

For the large distribution shift (bottom row), Deep Ensembles outperforms the other two methods by achieving much higher RQ values. This finding indicates that the majority of rejected observations are misclassified, effectively improving the NRA. MC Dropout has lower RQ values, with the NRA curve increasing slower. The standard NN falls quickly to RQ value 1.0, indicating that the uncertainty estimates are uninformative and rejections are random. This trend is reflected in the stagnating NRA.

7. Discussion

Methods with uncertainty estimation as XAI perform on par or better than the standard NN. If there is no shift, there is no difference between the methods. However, from the point a distribution shift is present, uncertainty estimation leads to optimal decision-making at smaller rejection rates. For small shifts, capturing model uncertainty induces higher initial accuracy and fewer rejections to obtain a specific level of accuracy. For large distribution shifts, it issues a warning about novel observations so the system can reject predictions accordingly, unlike the standard NN.

Uncertainty as XAI increases *trust* in an ML system by also capturing model uncertainty, indicating when an observation lies outside the observed training data. The results show that the model uncertainty values are sensitive to changes in the data distribution, providing an important source of information to the decision maker not available in the standard NN. Furthermore, *robustness* is improved as the total uncertainty grows with increasing distribution shifts, while uncertainty values in the standard NN decrease. Finally, *actionability* is increased by directly using the uncertainty information in a classification with rejection system, raising the NRA even under large distribution shifts.

Continuing on the increased actionability, the decision maker should inspect Figure 10 to decide on the appropriate rejection threshold given the specific

labeling budget and accuracy requirements. For example in the situation of no shift, it would be sensible to label at most 20% of the observations as the CQ and RQ decrease quickly, resulting in a slowly increasing NRA curve. In contrast, for the large shift, the full labeling budget can be used as the NRA continues to improve.

8. Conclusion

This paper proposes a general uncertainty framework positioning *uncertainty estimation in ML models as an XAI technique*, giving local and model-specific explanations. Furthermore, the framework uses *classification with rejection* to reduce misclassifications by bringing a human expert in the loop for uncertain observations. Finally, the framework is applied to a case study of *NNs* in educational data mining subject to distribution shifts.

The case study demonstrates that standard NNs only capturing data uncertainty are confidently wrong when confronted with distribution shifts. In contrast, NNs equipped with uncertainty estimation as XAI raise a warning in novel situations through increased model uncertainty, offering a solution to their overconfidence. Deep Ensembles outperform MC Dropout as an XAI technique with higher quality uncertainty estimates, obtaining higher accuracy when rejecting the most uncertain observations. Uncertainty as XAI improves the model's *trustworthiness* in downstream decision-making tasks, giving rise to more *actionable* and *robust* ML systems in OR.

Several directions for future work are possible. The case study only considers knowing the target labels in time due to limitations in the data; studies also satisfying the requirement for input features would help validate the findings. Although this paper focuses on uncertainty for NN classifiers, uncertainty can also be quantified for NN regression models and other ML models such as Gaussian Processes (Price et al., 2019) and Random Forests (Shaker & Hüllermeier, 2020). Finally, uncertainty as XAI can be used in active learning, where limited labeled training data is available and the ML system can ask a human expert to label the most uncertain observations (Kadziński & Ciomek, 2021).

Acknowledgments

This work was supported by the Research Foundation Flanders (FWO) [grant number 1S97022N].

References

Adadi, A., & Berrada, M. (2018). Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, *6*, 52138–52160. doi:10.1109/ACCESS.2018.2870052.

Bai, X., Wang, X., Liu, X., Liu, Q., Song, J., Sebe, N., & Kim, B. (2021). Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments. *Pattern Recognition*, *120*, 108102. doi:10.1016/j.patcog.2021.108102.

Barandas, M., Folgado, D., Santos, R., Simão, R., & Gamboa, H. (2022). Uncertainty-based rejection in machine learning: Implications for model development and interpretability. *Electronics*, *11*, 396. doi:10.3390/electronics11030396.

Bastos, J. A., & Matos, S. M. (2022). Explainable models of credit losses. *European Journal of Operational Research*, *301*, 386–394. doi:10.1016/j.ejor.2021.11.009.

Cabitza, F., Campagner, A., Malgieri, G., Natali, C., Schneeberger, D., Stoeger, K., & Holzinger, A. (2023). Quod erat demonstrandum?-towards a typology of the concept of explanation for the design of explainable ai. *Expert Systems with Applications*, *213*, 118888. doi:10.1016/j.eswa.2022.118888.

Choi, T.-M., Wallace, S. W., & Wang, Y. (2018). Big data analytics in operations management. *Production and Operations Management*, *27*, 1868–1883. doi:10.1111/poms.12838.

Condessa, F., Bioucas-Dias, J., & Kovačević, J. (2017). Performance measures for classification systems with rejection. *Pattern Recognition*, *63*, 437–450. doi:10.1016/j.patcog.2016.10.011.

Coussement, K., Phan, M., De Caigny, A., Benoit, D. F., & Raes, A. (2020). Predicting student dropout in subscription-based online learning environments: The beneficial impact of the logit leaf model. *Decision Support Systems*, *135*, 113325. doi:10.1016/j.dss.2020.113325.

De Caigny, A., Coussement, K., & De Bock, K. W. (2018). A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees. *European Journal of Operational Research*, *269*, 760–772. doi:10.1016/j.ejor.2018.02.009.

Deeva, G., De Smedt, J., Saint-Pierre, C., Weber, R., & De Weerdt, J. (2022). Predicting student performance using sequence classification with time-based windows. *Expert Systems with Applications*, *209*, 118182. doi:10.1016/j.eswa.2022.118182.

Delen, D., Topuz, K., & Eryarsoy, E. (2020). Development of a bayesian belief network-based dss for predicting and understanding freshmen student attrition. *European Journal of Operational Research*, *281*, 575–587. doi:10.1016/j.ejor.2019.03.037. Featured Cluster: Business Analytics: Defining the field and identifying a research agenda.

Delen, D., Zolbanin, H. M., Crosby, D., & Wright, D. (2021). To imprison or not to imprison: an analytics model for drug courts. *Annals of Operations Research*, *303*, 101–124. doi:10.1007/s10479-021-03984-7.

Depeweg, S., Hernandez-Lobato, J.-M., Doshi-Velez, F., & Udluft, S. (2018). Decomposition of uncertainty in bayesian deep learning for efficient and risk-sensitive learning. In *International Conference on Machine Learning* (pp. 1184–1193). PMLR.

Der Kiureghian, A., & Ditlevsen, O. (2009). Aleatory or epistemic? does it matter? *Structural safety*, *31*, 105–112. doi:10.1016/j.strusafe.2008.06.020.

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, . doi:10.48550/arXiv.1702.08608.

Feng, R., Ma, A., Jing, Z., Gu, X., Dang, P., & Yao, B. (2022). Understanding the uncertainty of traffic time prediction impacts on parking lot reservation in logistics centers. *Annals of Operations Research*, (pp. 1–23). doi:10.1007/s10479-022-04734-z.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning* (pp. 1050–1059). PMLR.

Gardner, J., & Brooks, C. (2018). Student success prediction in moocs. *User Modeling and User-Adapted Interaction*, *28*, 127–203. doi:10.1007/s11257-018-9203-z.

Garvey, M. D., Carnovale, S., & Yeniyurt, S. (2015). An analytical framework for supply network risk propagation: A bayesian network approach. *European Journal of Operational Research*, *243*, 618–627. doi:10.1016/j.ejor.2014.10.034.

Gašević, D., Dawson, S., Rogers, T., & Gasevic, D. (2016). Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting academic success. *The Internet and Higher Education*, *28*, 68–84. doi:10.1016/j.iheduc.2015.10.002.

Ghahtarani, A. (2021). A new portfolio selection problem in bubble condition under uncertainty: Application of z-number theory and fuzzy neural network. *Expert Systems with Applications*, *177*, 114944. doi:10.1016/j.eswa.2021.114944.

Gunnarsson, B. R., vanden Broucke, S., Baesens, B., Óskarsdóttir, M., & Lemahieu, W. (2021). Deep learning for credit scoring: Do or don't? *European Journal of Operational Research*, *295*, 292–305. doi:10.1016/j.ejor.2021.03.006.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International Conference on Machine Learning* (pp. 1321–1330). PMLR.

Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, *110*, 457–506. doi:10.1007/s10994-021-05946-3.

Huyen, C. (2022). Designing machine learning systems. chapter 8. O'Reilly Media, Inc.

Kadziński, M., & Ciomek, K. (2021). Active learning strategies for interactive elicitation of assignment examples for threshold-based multiple criteria sorting. *European Journal of Operational Research*, *293*, 658–680. doi:10.1016/j.ejor.2020.12.055.

Kraus, M., & Feuerriegel, S. (2019). Forecasting remaining useful life: Interpretable deep learning approach via variational bayesian inferences. *Decision Support Systems*, *125*, 113100. doi:10.1016/j.dss.2019.113100.

Kraus, M., Feuerriegel, S., & Oztekin, A. (2020). Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research*, *281*, 628–641. doi:10.1016/j.ejor.2019.09.018. Featured Cluster: Business Analytics: Defining the field and identifying a research agenda.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, *30*.

Mena, J., Pujol, O., & Vitrià, J. (2021). A survey on uncertainty estimation in deep learning classification systems from a bayesian perspective. *ACM Computing Surveys (CSUR)*, *54*, 1–35. doi:10.1145/3477140.

[dataset] MITx, & HarvardX (2014). Hmxpc13_di_v2_5-14-14.csv. In *HarvardX-MITx Person-Course Academic Year 2013 De-Identified dataset, version 2.0*. Harvard Dataverse. doi:10.7910/DVN/26147/OCLJIV.

Murphy, K. (2022). Probabilistic machine learning: Advanced topics. chapter 20. MIT Press.

Nahta, R., Meena, Y. K., Gopalani, D., & Chauhan, G. S. (2021). A hybrid neural variational cf-nade for collaborative filtering using abstraction and generation. *Expert Systems with Applications*, *179*, 115047. doi:10.1016/j.eswa.2021.115047.

Oh, B., Hwang, J., Seo, S., Chun, S., & Lee, K.-H. (2021). Inductive gaussian representation of user-specific information for personalized stress-level prediction. *Expert Systems with Applications*, *178*, 114912. doi:10.1016/j.eswa.2021.114912.

Olaya, D., Vásquez, J., Maldonado, S., Miranda, J., & Verbeke, W. (2020). Uplift modeling for preventing student dropout in higher education. *Decision Support Systems*, *134*, 113320. doi:10.1016/j.dss.2020.113320.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., & Snoek, J. (2019). Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, *32*.

Phan, M., De Caigny, A., & Coussement, K. (2023). A decision support framework to incorporate textual data for early student dropout prediction in higher education. *Decision Support Systems*, (p. 113940). doi:10.1016/j.dss.2023.113940.

Piri, S., Delen, D., Liu, T., & Zolbanin, H. M. (2017). A data analytics approach to building a clinical decision support system for diabetic retinopathy: Developing and deploying a model ensemble. *Decision Support Systems*, *101*, 12–27. doi:10.1016/j.dss.2017.05.012.

Price, I., Fowkes, J., & Hopman, D. (2019). Gaussian processes for unconstraining demand. *European Journal of Operational Research*, *275*, 621–634. doi:10.1016/j.ejor.2018.11.065.

Sachan, S., Yang, J.-B., Xu, D.-L., Benavides, D. E., & Li, Y. (2020). An explainable ai decision-support-system to automate loan underwriting. *Expert Systems with Applications*, *144*, 113100. doi:10.1016/j.eswa.2019.113100.

Shaker, M. H., & Hüllermeier, E. (2020). Aleatoric and epistemic uncertainty with random forests. In *Advances in Intelligent Data Analysis XVIII: 18th International Symposium on Intelligent Data Analysis, IDA 2020, Konstanz, Germany, April 27–29, 2020, Proceedings 18* (pp. 444–456). Springer. doi:10.1007/978-3-030-44584-3_35.

Shin, D. (2021). The effects of explainability and causability on perception, trust, and acceptance: Implications for explainable ai. *International Journal of Human-Computer Studies*, *146*, 102551. doi:10.1016/j.ijhcs.2020.102551.

Van Belle, J., Guns, T., & Verbeke, W. (2021). Using shared sell-through data to forecast wholesaler demand in multi-echelon supply chains. *European Journal of Operational Research*, *288*, 466–479. doi:10.1016/j.ejor.2020.05.059.

Van Nguyen, T., Zhou, L., Chong, A. Y. L., Li, B., & Pu, X. (2020). Predicting customer demand for remanufactured products: A data-mining approach. *European Journal of Operational Research*, *281*, 543–558. doi:10.1016/j.ejor.2019.08.015.

Verboven, S., Berrevoets, J., Wuytens, C., Baesens, B., & Verbeke, W. (2021). Autoencoders for strategic decision support. *Decision Support Systems*, *150*, 113422. doi:10.1016/j.dss.2020.113422. Interpretable Data Science For Decision Making.

Weytjens, H., & De Weerdt, J. (2022). Learning uncertainty with artificial neural networks for predictive process monitoring. *Applied Soft Computing*, *125*, 109134. doi:10.1016/j.asoc.2022.109134.

Whitehill, J., Mohan, K., Seaton, D., Rosen, Y., & Tingley, D. (2017). Mooc dropout prediction: How to measure accuracy? In *Proceedings of the fourth (2017) acm conference on learning@ scale* (pp. 161–164). doi:10.1145/3051457.3053974.

Wilson, A. G. (2020). The case for bayesian deep learning. *arXiv preprint arXiv:2001.10995*, . doi:10.48550/arXiv.2001.10995.

Yong, B. X., & Brintrup, A. (2022). Bayesian autoencoders with uncertainty quantification: Towards trustworthy anomaly detection. *Expert Systems with Applications*, *209*, 118196. doi:10.1016/j.eswa.2022.118196.

Yu, J., Alrajhi, L., Harit, A., Sun, Z., Cristea, A. I., & Shi, L. (2021). Exploring bayesian deep learning for urgent instructor intervention need in mooc forums. In *International Conference on Intelligent Tutoring Systems* (pp. 78–90). Springer. doi:10.1007/978-3-030-80421-3_10.

Zhang, X., & Mahadevan, S. (2020). Bayesian neural networks for flight trajectory prediction and safety assessment. *Decision Support Systems*, *131*, 113246. doi:10.1016/j.dss.2020.113246.