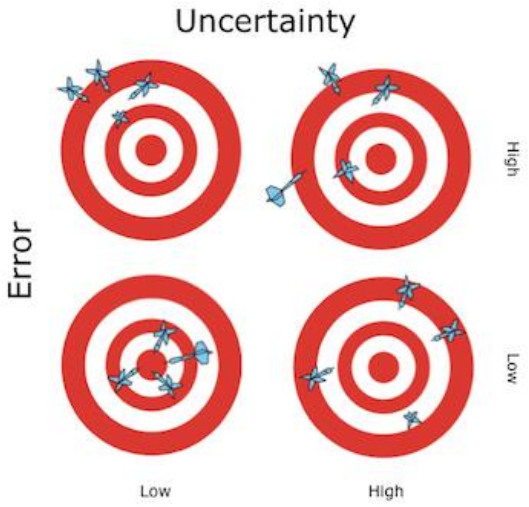
UNIVERSITÉ DU LUXEMBOURG

Probabilistic Machine Learning

What is Uncertainty?

Environmental systems are inherently complex, characterized by variability in natural processes, human activities, and numerous unknown aspects. This inherent unpredictability is referred to as **uncertainty**. It is important to note that uncertainty does not imply **error**; rather, it reflects the limitations of our knowledge or the natural variability of a system.

In contrast, **error** refers to the difference between a measured, predicted, or observed value and the true or accepted value. **Calculating error requires knowledge or an assumption of the true value**. Without this reference point, error cannot be directly measured. When the true value is unknown, we cannot discuss "error" directly; instead, we address **uncertainty**, which acknowledges the lack of a definitive reference.



Uncertainty vs. Error.

Uncertainty is often classified into two types:

1. Aleatory Uncertainty

Also known as stochastic or irreducible uncertainty, it arises from the inherent randomness or variability of a system. This type of uncertainty is often considered irreducible because it stems from natural variability or chaotic processes that cannot be eliminated, even with perfect knowledge or measurements. An example is weather

variabilit**y,** Even with precise models, the chaotic nature of weather systems means there is always some unpredictability in long-term forecasts.

2. Epistemic Uncertainty

Also known as reducible or knowledge-based uncertainty, it arises from a lack of understanding, incomplete information, or insufficient data. Unlike aleatory uncertainty, epistemic uncertainty can be reduced by improving models, collecting more data, or gaining better insights into the underlying processes. An example is uncertainty in mapping underground resources due to limited sampling points, which could be reduced with more comprehensive measurements.

Note that while Aleatory uncertainty and epistemic uncertainty are conceptual distinctions, in practice, the boundary between them can blur. *What is considered random (aleatory) today may later be understood as a predictable process (epistemic) once more knowledge or better models are developed. This is why some argue that aleatory uncertainty could be considered a subset of epistemic uncertainty—if we truly knew all the underlying processes, the randomness might diminish or disappear. For example, the flipping of a coin is often described as aleatory uncertainty because the outcome appears inherently random. However, with precise knowledge of the coin's weight, spin velocity, and air resistance, we might predict the result, converting aleatory uncertainty into epistemic uncertainty.*

Aleatory uncertainty often represents processes that appear random ***given our current understanding****. With deeper insights, some aleatory aspects could shift into epistemic territory.*

Classification of Uncertainty

Another relevant categorization of uncertainty, especially in the context of environmental data, is based on **where the uncertainty arises in the data-analysis process**. This categorization includes the following types:

1. **Data Uncertainty:** Refers to uncertainty originating from the input data used for analysis or modeling. It arises due to errors or limitations in the instruments or sensors used to collect data, missing data, or inconsistencies in datasets.
2. **Representativeness Uncertainty:** Results from the inability of data to accurately represent the actual characteristics of the phenomenon it was intended to describe. This occurs when the spatial, temporal, or contextual coverage of the data is inadequate. *These are some examples: Using data from a single groundwater well to represent an entire aquifer or relying on monthly averages that mask daily variability.*
3. **Structural Uncertainty**: Arises from limitations or simplifications in the conceptual framework of a model or analysis. It reflects uncertainty in how the system is represented. *These are some examples: Neglecting certain key input*

features (e.g., omitting land-use changes in a climate model); Assuming that a system is linear when it is inherently nonlinear (e.g., river discharge relationships in hydrological models).

4. **Parameter Uncertainty**: Refers to uncertainty in the values of parameters used in models due to limited data, natural variability, or inaccuracies. Unlike data uncertainty, **parameters are not directly measured** but inferred from expert knowledge, literature, or general understanding.
5. **Scenario Uncertainty:** Reflects the uncertainty in selecting or assuming future scenarios (e.g., estimating the future population of a city).

When modeling environmental systems, uncertainties from various sources **propagate** through the model, influencing outcomes in different ways:

6. **Model Uncertainty**: When simulating the past state of an environmental system (e.g., reconstructing historical climate conditions), structural, parameter, and data uncertainties collectively propagate through the model and result in model uncertainty.
7. **Predictive Uncertainty:** When predicting the future state of an environmental system, factors contributing to model uncertainty, combined with scenario uncertainty, collectively propagate through the model, resulting in predictive uncertainty.

Quantifying Uncertainty

Probability is a widely used mathematical framework to quantify uncertainty. In this context, instead of assigning a single fixed value, the uncertain quantity is described using a range of possible values. The two types of uncertainty are viewed differently in terms of how they are quantified:

- **Epistemic Uncertainty:** This type of uncertainty, arising from a lack of knowledge, is often addressed using Bayesian methods. In Bayesian inference, probabilities represent degrees of belief or confidence about unknown quantities (e.g., model parameters) and are updated as more data becomes available.
- **Aleatory Uncertainty:** This type of uncertainty, inherent to the variability of the system, is typically quantified using frequentist approaches or stochastic modeling, where probabilities represent the long-term frequencies of outcomes under repeated trials or inherent randomness in the system.

Despite these differences in interpretation and approach, both epistemic and aleatory uncertainties ultimately converge to the same concept of quantification using probability distributions. This allows for a unified representation of uncertainty, whether it originates from incomplete knowledge or intrinsic randomness.

Note: *Some have argued that* ***fuzzy logic*** *provides a better representation of epistemic uncertainty compared to probabilistic methods. Fuzzy logic represents uncertainty through degrees of membership in fuzzy sets rather than probability distributions, making it particularly suitable for handling imprecise or ambiguous information, such as linguistic descriptions ("high temperature" or "low risk"). This approach can model uncertainty arising from incomplete knowledge without requiring precise probabilities, offering an alternative perspective for certain applications. However, we will not delve deeper into this concept here.*

The probabilistic quantification of uncertainty is approached differently depending on the type of uncertainty in environmental data analytics (and other fields). When a computational or predictive model is not used to describe the system, uncertainty is estimated using methods such as **statistical analysis** of data or expert judgment. Conversely, when quantifying model or predictive uncertainty, it is often necessary to first quantify input uncertainties and then perform **uncertainty propagation analysis** to evaluate their combined effects on model outputs. Alternatively, probabilistic models, such as those in **probabilistic machine learning**, can directly incorporate and represent uncertainties in predictions.

Quantifying Uncertainty Through Statistical Analysis

A. When True Values Are Known

When true values (or at least a reliable benchmark) are known or available, statistical analysis can directly quantify uncertainty by comparing observed values to the true values. This is particularly relevant in remote sensing, where, for example, satellite-measured land surface temperature can be compared with ground-truth (field) measurements to calculate errors. Field measurements are often assumed to have significantly lower error than remote sensing data, and their uncertainty is typically ignored, allowing them to serve as a proxy for the true values.

The process involves the following steps:

1) Compute the error as the difference between the observed values (e.g., satellite measurements) and the true values (e.g., field measurements)

2) Evaluate the residuals (calculated errors) to derive a distribution of errors. This distribution reflects the uncertainty in the observed data.

B. When True Values Are Not Available

When true values are unavailable, statistical analysis can capture only variability, which is often used as a **proxy** for uncertainty. However, it is important to note that variability is not the same as uncertainty. Variability reflects inherent randomness (**aleatory**

uncertainty) but may fail to account for systematic biases or gaps in knowledge (**epistemic uncertainty**).

For example, in a rainfall dataset, variability across regions or seasons can be analyzed to estimate uncertainty in annual rainfall averages. However, this approach does not address potential biases in data collection methods.

Resampling methods like **bootstrapping** are powerful statistical tools to approximate uncertainty when true values are unavailable.

This is how bootstrapping works:

1) Start with a dataset containing n observations (e.g., daily rainfall measurements over a year). *Example:* $X = \{10,15,12,9,13,14,11\}$.

2) Randomly draw samples **with replacement** from the original dataset to create a "resampled dataset" of the same size (n). Each resample can include duplicate data points. *Example: A resampled dataset might look like* $X' = \{15,12,12,10,14,10,13\}$.

3) Generate many such resampled datasets (e.g., 1,000 or more) to simulate the variability in the data.

4) Compute the statistic of interest (e.g., mean, variance, median) for each resampled dataset.

5) Analyze the distribution of the calculated statistics (e.g., the mean values across all resamples) to estimate the **uncertainty** in the statistic. *Example: Create a histogram of the mean rainfall values from the 1,000 resamples to visualize variability.*

Uncertainty Propagation Analysis

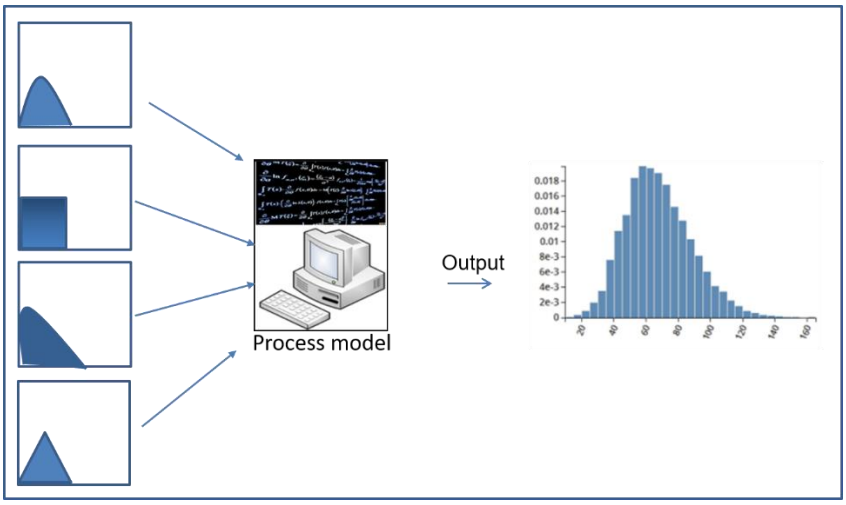
When using models (whether physics-based or data-driven ML models), uncertainty in the inputs will propagate to the outputs, a phenomenon commonly referred to as **uncertainty propagation**. The process of analyzing how previously quantified input uncertainty affects the outputs in a quantitative way is known as **uncertainty propagation analysis** (UPA).

UPA typically involves **deterministic models** that do not inherently account for uncertainty. The analysis evaluates how variations in input parameters influence the model's predictions.

By far, the most widely used method for UPA is **Monte Carlo simulation** (MCS). In MCS the process begins by defining probability distributions for uncertain input variables. Monte Carlo then generates many random samples from these input distributions (commonly using simple random sampling, or Latin hypercube sampling) and feeds them into the model. For each sample, the model computes an output, creating a distribution of outputs that reflects how input uncertainties propagate through the model. By analyzing the output distribution (e.g., mean, variance, confidence intervals),

Monte Carlo provides insights into the uncertainty of predictions. This approach is particularly valuable for evaluating the robustness of ML models in real-world scenarios where input data is inherently uncertain.

In ML, MCSs are typically performed after training, using the trained model to make predictions repeatedly. These simulations often involve generating many predictions (in the order of 10,000 repetitions or more, depending on the complexity of the model and the level of precision required) by sampling from input distributions, model parameters, or noise to assess the variability and uncertainty in the model's outputs.



Monte Carlo Analysis.

Probabilistic Models

Probabilistic models do not require an external algorithm like Monte Carlo to quantify uncertainty in model outputs. These models inherently account for uncertainty by representing inputs, outputs, and model parameters as probability distributions rather than fixed values, allowing uncertainty to be modeled and propagated within the framework of the model itself.

Here, we focus specifically on probabilistic models within the context of machine learning.

Probabilistic Machine Learning (Probabilistic ML) is a branch of machine learning that explicitly incorporates probability theory to model and reason about uncertainty in data and predictions. Instead of making deterministic predictions (as common ML models do), **probabilistic ML approaches output probabilities or distributions. Bayesian ML models** are a cornerstone of **probabilistic ML**, as they explicitly use probability theory to model uncertainty in data and parameters.

Note: *When epistemic uncertainty is quantified using fuzzy logic,* ***Fuzzy Neural Networks (FNNs)*** *become relevant as an alternative to probabilistic machine learning models. FNNs integrate the learning capabilities of neural networks with the interpretability of fuzzy logic, representing uncertainty through* ***membership functions*** *and* ***fuzzy rules*** *rather than probability distributions. This approach captures the degree of vagueness or imprecision in inputs and outputs, making FNNs particularly useful for systems where knowledge is incomplete or linguistic descriptions (e.g., "low risk" or "high temperature") are prevalent. Unlike probabilistic models, which rely on probability distributions, FNNs use fuzzy reasoning to manage uncertainty, offering a rule-based and interpretable framework.*

Bayesian ML Models

The key notion behind **Bayesian ML models** is the use of **Bayes' theorem** to update and refine our beliefs about model parameters or predictions as new data becomes available. Bayesian ML models treat unknown quantities (parameters, predictions, etc.) as **random variables** and model them using probability distributions, allowing for uncertainty quantification and the integration of prior knowledge.

Reminder:

Bayes' Theorem: Bayesian models are based on this fundamental formula:

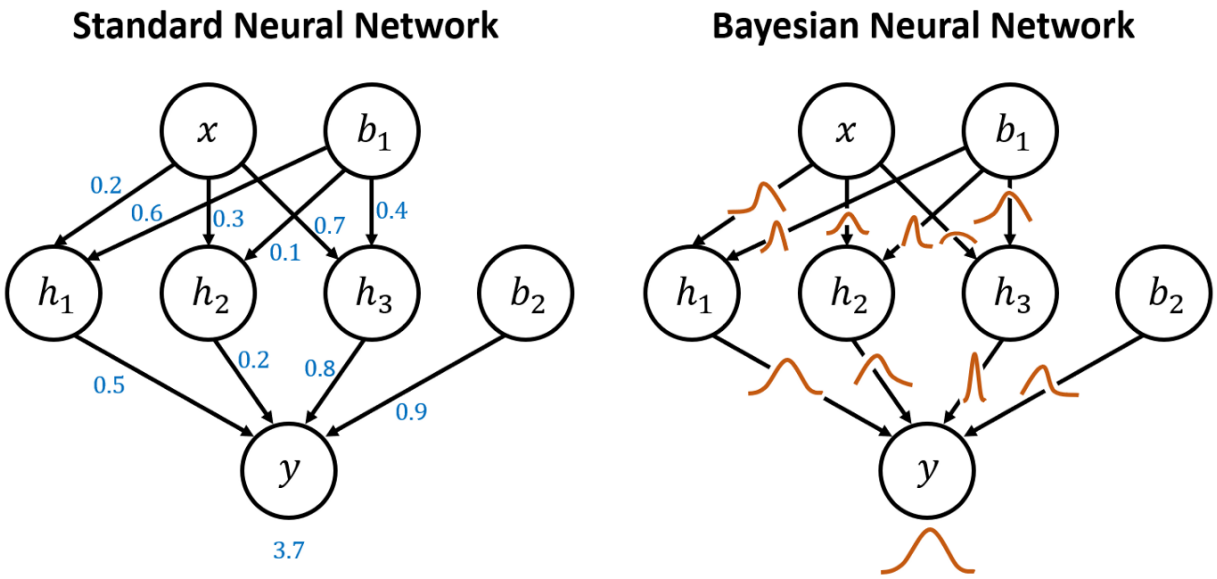
$$P(\text{parameters}|\text{data}) = \frac{P(\text{data}|\text{parameters}) \cdot P(\text{parameters})}{P(\text{data})}$$

- **Posterior** ($P(\text{parameters}|\text{data})$): Updated belief about parameters after observing data.
- **Likelihood** ($P(\text{data}|\text{parameters})$): Probability of observing the data given the parameters.
- **Prior** ($P(\text{parameters})$): Initial belief about the parameters before observing data.
- **Evidence** ($P(\text{data})$): Normalizing constant to ensure the probabilities sum to 1.

There are several types of Bayesian ML models, including **Gaussian Processes (GPs)**, **Bayesian Neural Networks (BNNs)**, and **Variational Autoencoders (VAEs)**. In the following, I will focus on **BNNs**, a widely used and popular approach in modern ML, to explain the concept.

In essence, BNNs are typical Neural Networks (of almost any architecture like FFNNs, CNNs, RNNs, etc.) but differ in how weights and biases are treated, trained, and used for predictions. In BNNs, weights and biases are modeled as probability distributions

instead of fixed values. During inference, for a given input, multiple forward passes are performed with different sampled weights and biases. Each pass produces a slightly different prediction due to the randomness in weights and biases. These multiple predictions are aggregated to form a **probabilistic output**.



Bayesian Neural Networks.

BNNs use backpropagation for training, but with key differences from typical neural networks. At the start of training, weights and biases are assigned **prior** probability distributions, often Gaussian, based on assumptions or prior knowledge. Typically, all weights and biases start with the same type of prior distribution, but the parameters of these priors (e.g., mean and variance) can vary depending on domain knowledge or initialization strategy. For instance, if you expect some weights to have smaller magnitudes, you might set their prior variance to a smaller value.

As training progresses, these distributions are updated using Bayesian inference (Bayes' theorem), integrating prior knowledge with the likelihood of the data. The posterior distributions of weights and biases are estimated iteratively, often using **Variational inference**.

Variational inference

Instead of directly computing the posterior $P(w \mid D)$, which is intractable for neural networks, Variational inference introduces an approximate posterior $Q(w;\theta)$ (a simplified version of the posterior, chosen to make the problem tractable), with its own parameters θ (e.g., mean (μ) and variance (σ^2) for a Gaussian distribution). In each step

of training, a set of weight and bias samples (w) is drawn from the approximate posterior $Q(w;\theta)$ using the current parameters $\theta = \{\mu, \sigma^2\}$ (if Gaussian). Each sampled combination of weights and biases (w) defines a specific instance of the BNN. For each sample of w, the network performs a **forward pass** over the training data to compute the output, and multiple forward passes (one per sample of w) result in a **distribution of outputs** for the given input data. The **likelihood** of the data given these weights is then computed as:

$$\log P(\mathcal{D}|w) = \sum_{i=1}^{N} \log P(y_i|\mathbf{x}_i, w)$$

Where y_i is the **target value** (what the model is trying to predict, for example the correct label in a classification task), x_i is the **input features** (the data used for prediction), w is the **weights and biases** of the neural network, and $P(y_i|x_i, w)$ is the **predicted probability** of y_i being the correct label or value, given the input x_i and the specific weights w.

The likelihood $P(D \mid w)$ (data fit) and the prior $P(w)$ together inform the objective function:

$$\mathcal{L}(\theta) = \mathbb{E}_{Q(w;\theta)}[\log P(\mathcal{D}|w)] + \mathbb{E}_{Q(w;\theta)}[\log P(w)]$$

Finally, Backpropagation is used to optimize θ.

BNNs are very useful in environmental data analytics, given the inherent uncertainty and complexity in environmental data and systems. BNN allows providing probabilistic predictions with confidence intervals, which are critical for risk assessment and decision-making in environmental data analytics. They are commonly used in probabilistic modeling of temperature, rainfall, or extreme weather events, flood risk assessments, predicting species population dynamics, among many other.

However, BNNs are computationally expensive to train (As multiple forward passes are required for each training step), which often limits the size of the model. This limitation is a challenge both generally and in Environmental Data Analytics.