Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2025) 549-568

ECCOMAS
Proceedia

UNCECOMP 2025
6th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Rhodes Island, Greece, 15-18 June 2025

# INPUT PARAMETER SPECIFIC UNCERTAINTY QUANTIFICATION OF DEEP LEARNING MODELS USING MONTE CARLO DROPOUT

**Eugen Boos[1,*], Maik Linnemann[2], Hajo Wiemer[1], Verena Psyk[2] and Steffen Ihlenfeldt[1,2]**

[1] Institute of Mechatronic Engineering, Technische Universität Dresden, Germany

[2] Fraunhofer Institute for Machine Tools and Forming Technology, Chemnitz, Germany

* e-mail: eugen.boos@tu-dresden.de

## Abstract

*Low-data environments, common in industrial use-cases, exhibit significantly elevated levels of both model and data uncertainty in deep learning applications. Data generation in such settings is generally financially expensive. Therefore, strategic data collection plays a critical role in improving the performance of deep learning models. Effective model uncertainty quantification plays an essential role in identifying input domains with poor model confidence, enabling more efficient data collection strategies. While Monte Carlo dropout is a well-known method for model uncertainty quantification, this study explores a novel input parameter specific application of Monte Carlo dropout to sequentially applied input channel layers rather than the entire model. This approach allows for a more detailed uncertainty quantification focused on specific input parameters, yielding additional insights into model robustness and model confidence across different input domains. A case study on condition monitoring in manufacturing using the Temporal Fusion Transformer demonstrates how these insights can guide smarter data generation strategies, ultimately improving model performance in low-data, high-cost environments.*

**Keywords:** Uncertainty Quantification, Monte Carlo Dropout, Deep Learning, Design of Experiment

ISSN:2623-3339 
Peer-review under responsibility of the organizing committee of UNCECOMP 2025.
doi: 10.7712/120225.12377.20964

# 1 INTRODUCTION

Machine Learning (ML) and particularly Deep Learning (DL) have seen a significant growth in development as well as application in various engineering disciplines, particularly speaking in production engineering. These developments lead to meaningful progress in areas of application such as for instance condition monitoring [1] and predictive maintenance [2]. The resulting advancements enable improved efficiency, reduced downtime, and enhanced decision-making through data-driven insights. However, despite these advancements, the application of ML and DL in production engineering environments still presents unique challenges. One of the most significant limitations in engineering environments are low data quantities and high generation costs [3]. Especially in regards to data collected from malfunctioning processes or machines. In contrast to other domains, where large datasets are accessible, engineering applications often cope with data that is sparse, noisy, and costly to generate. These constraints hinder the ability of data-driven models to repeatedly output reliable predictions. This is particularly relevant for DL models, as they are more dependent on large datasets while generally achieving a higher accuracy than traditional ML models [4].

A critical aspect influencing the performance and reliability of DL models in these contexts is uncertainty. Uncertainty impacts model explainability, robustness and overall trustworthiness [5]. Uncertainty quantification (UQ) addresses these factors, resulting in an overall improved insight into model performance and reliability [6]. This is particularly the case in low-data engineering environments [7]. Uncertainty can be categorized into epistemic and aleatoric uncertainty, both of which influence model predictions differently [8], [9]. Epistemic uncertainty arises from a lack of knowledge about the model parameters and can be reduced by incorporating additional data, improving model architecture or refining the parameters underlying assumptions. This type of uncertainty is particularly relevant in cases where the model has not encountered sufficient examples of a specific input, making it essential for identifying out-of-distribution samples. In contrast, aleatoric uncertainty is inherent to the input data itself and cannot be mitigated through additional observations. It results from stochastic noise in the input, such as sensor errors, occlusion or environmental variations. Unlike epistemic uncertainty, which reflects model limitations, aleatoric uncertainty characterizes the inherent randomness in the system being modeled itself. While epistemic uncertainty can be directly addressed through data augmentation or improved modeling strategies, aleatoric uncertainty must be accounted for probabilistically to ensure robust and reliable predictions in real-world applications.

Generally, increasing the amount of available data reduces model uncertainty [7]. However, in environments where data generation is costly, strategic data generation and selection become crucial to efficiently improve model confidence while minimizing resource expenditure. Traditional UQ methods primarily focus on characterizing the uncertainty of the entire model, making them more effective in capturing epistemic uncertainty [6, 10, 11]. These approaches, while valuable for quantifying overall model uncertainty, do not offer a deeper look into the underlying sources of the uncertainty. Thus, besides general improvement strategies, such as increasing data quantity or improved model architectures, a strategic design of experiments (DoE) to cost efficiently lower model uncertainty is not directly derivable from UQ. Enabling an input parameter-specific UQ allows for a more specific identification of uncertainty sources, specifically in regards to the lack of data in smaller sized data sets. This approach enhances uncertainty interpretability and explainability by addressing the contribution of individual input parameters onto the overall model uncertainty, thereby distinguishing between inherent variability in the data and uncertainty stemming from model limitations. By isolating these two,

targeted mitigation strategies can be developed, optimizing DoE and resource allocation for improving model robustness and predictive reliability.

### 1.1 Research objectives and main contribution

Leveraging advanced DL models and UQ techniques becomes crucial in improving model reliability, optimizing financial resource allocation and enhancing predictive performance despite data scarcity. This study aims to address these challenges by developing an input parameter-specific UQ approach based on Monte Carlo dropout (MCD) [12]. By further distinguishing between aleatoric and epistemic uncertainty, this method enhances model interpretability and supports a strategic DoE to cost efficiently reduce uncertainty. Although aleatoric uncertainty remains constant in big data environments, since it is tied to intrinsic noise in the data rather than data scarcity [7], in small-data environments, its quantification improves as more data becomes available [13]. With limited data, aleatoric uncertainty estimates are often biased due to insufficient representation of the underlying stochastic variations. As the dataset grows, the model gains a more accurate understanding of the inherent noise, leading to a more reliable uncertainty quantification and an improved predictive performance. This underscores the importance of strategic data generation in small-data environments to enhance the robustness of uncertainty estimation. For demonstrating and validating the proposed method, this study employs experimental data from the electromagnetic forming (EMF) process [14].

## 2 RELATED WORK

### 2.1 Methods for uncertainty estimation in deep learning

UQ in DL has gained increasing attention as models are deployed in applications requiring reliable decision-making. While traditional research has primarily focused on improving point-estimate prediction accuracy, recent review studies show a growing tendency towards UQ [6], [15]. The quantification of uncertainty in deep neural networks can be roughly categorized into four groups: deterministic models, Bayesian methods, ensemble methods and test-time augmentation methods.

Deterministic methods estimate uncertainty by providing a scalar value or measure that quantifies the level of uncertainty associated with a prediction. These methods assume that the underlying model is deterministic and that uncertainty can be inferred from a single forward pass through the network, rather than requiring multiple stochastic evaluations. Prevalent methods are Evidentuial Deep Learning [16], Prior Networks [17] and distance aware networks [18]. Additionally, uncertainty-aware loss functions, such as quantile loss [19], are also part of deterministic models. Their primary advantage lies in their computational efficiency and simplicity, making them well-suited for applications requiring rapid inference with minimal overhead. Bayesian methods or Bayesian Neural Networks (BNN) are a class of approaches that incorporate uncertainty by representing model parameters as probabilistic distributions rather than fixed values [20]. This framework enables Bayesian methods to estimate predictive uncertainty by capturing the distribution of possible outputs for a given input, rather than a single-point prediction. However, computing the exact posterior distribution is intractable for deep neural networks, requiring the use of approximation methods [9]. The most prevailing approximation methods are Variational Inference (VI) [21] and Monte Carlo dropout (MCD) [12, 22]. VI approximates the posterior by optimizing a simpler normal distribution with the assumption the input data follows a normal distribution, while MCD estimates uncertainty by applying dropout at inference time, effectively sampling from different subnetworks to generate a distribution of predictions. Bayesian methods provide robust frameworks for quantifying both

epistemic and aleatoric uncertainty. However, their computational complexity is high, and their scalability remains limited compared to alternative approaches. Quite similar to the concept of sampling from multiple sub-networks are deep ensemble networks (DE). They estimate uncertainty by leveraging the output diversity among multiple independent neural networks for the same input to capture epistemic uncertainty [23]. Trained with different weight initializations or dataset partitions DE networks learn diverse representations of the data, allowing the ensemble to approximate the posterior distribution over model predictions. By aggregating the outputs through methods such as averaging, DE reduces overconfidence and provides a more reliable measure of epistemic uncertainty, particularly in cases where data is sparse or out-of-distribution. However, they are computationally expensive, scale inefficiently due to multiple forward passes and do not explicitly model aleatoric uncertainty. Test-Time Augmentation methods produce uncertainty estimations by applying data augmentations to input samples during inference, generating multiple predictions per input instead of a single deterministic output [24]. By analyzing the variance among augmented predictions, Test-Time Augmentation provides a computationally efficient estimation on aleatoric uncertainty, without requiring modifications to model architecture or training procedures. It is however mainly applicable in computer vision [6, 15].

### 2.2 Methods for strategic data generation and design of experiment

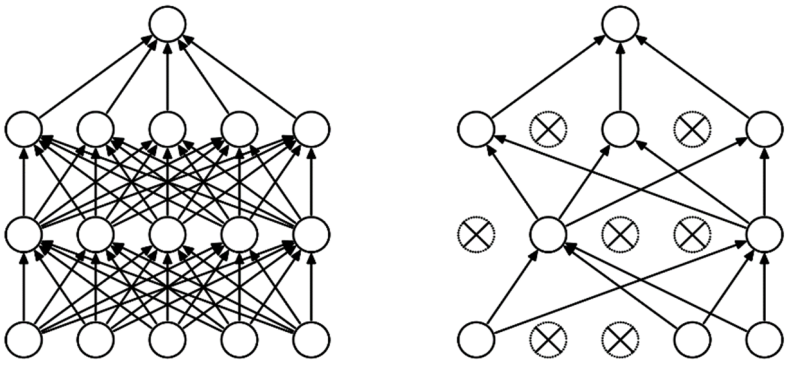
DoE is a significant tool in sparse data environments to strategically generate real world data for data driven applications, such as ML and DL. As often financial [25] or time [26] restrictions limit data generation, using the given resources strategically can influence the underlying model performance significantly. To address these hurdles there are different approaches. One of the most straight forward strategics for DoE comes down to space-filling and quasi-random sampling methods. They ensure uniform coverage of the input space by distributing samples evenly across the design space, making them ideal as an initial DoE strategy when data and model uncertainty is yet unknown. Techniques such as Latin Hypercube Sampling, Sobol Sequences, Halton Sequences, and Maximin Sampling help minimize gaps in the data space [27]. Space-filling methods offer a fixed experimental design for yet unknown data environments, reduce the risk of overfitting and contribute to model robustness. However, they do not account for aleatoric or epistemic uncertainty as well as model performance. In contrast, data-centric adaptive and sequential sampling strategies are designed to iteratively refine the data selection process to improve model performance while efficiently reducing uncertainty. Unlike fixed space-filling experimental designs, these methods dynamically adapt data collection based on model feedback, prioritizing high-uncertainty or high-value samples for learning. Well known approaches include active learning [28] and reinforcement learning-based adaptive sampling [29]. These strategies are particularly effective in environments, where data is expensive to obtain and the goal is to achieve maximum learning with minimal labeled data. As well as preferably environments where data generation and model training are directly intertwined enabling an automated or semi-automated training process. Alternatively, model-centric approaches focus on optimizing data generation through a probabilistic framework that systematically reduces model uncertainty. One of the most prevalent is Bayesian Experimental Design (BED) [30, 31] which selects experiments to maximize expected information gain, ensuring that each generate data point contributes maximally to refining posterior distributions, thus reducing epistemic uncertainty. By leveraging Bayesian inference, BED focuses on identifying the most informative regions in the design space. This makes it particularly effective for tasks where data collection is costly or limited. While BED ensures optimal data collection, it may be less practical for real-time applications, due to its high computational cost, requiring extensive posterior inference or optimization of expected information gain. In contrast to data-

and model-centric approaches, uncertainty-driven DoE methods target data generation by prioritizing regions of high uncertainty in the input space. These methods aim to reduce both epistemic and aleatoric uncertainty by selecting samples that are most informative for improving model predictions. Unlike data-centric approaches, which aim for multiple objectives such as sample diversity and model performance, uncertainty-driven methods focus mainly on reducing predictive uncertainty. Common techniques include variance-based sampling [32], which identifies regions with high predictive variance, and entropy-based sampling [33], which selects data points with high uncertainty across possible outputs. Additionally, approaches like Query-By-Committee leverage disagreements among ensemble models to guide sample selection, ensuring diverse and informative data collection [21]. By dynamically adapting the sampling process based on uncertainty feedback, these methods are particularly effective in environments where labeled data is scarce or expensive, enabling efficient learning with minimal data generation while enhancing model robustness and generalization.

## 3 FUNDAMENTELS

### 3.1 Dropout

Dropout is a well-known standard regularization practice [34] used to lessen overfitting in deep neural networks. During training, dropout prevents the model from over-relying on specific neurons by stochastically deactivating a subset of neurons in each iteration, thereby enforcing a more distributed representation of the input data. However, during inference, all neurons remain active, with their outputs are scaled by the dropout rate to ensure consistency between training and testing. The dropout rate governs the probability of neuron deactivation and is advised to be numerically optimized based on model architecture and dataset characteristics to achieve optimal generalization performance. Dropout is generally applied independently to each layer. Figure 1(b) illustrates the effect of applying dropout to a neural network during training in comparison to one without the application of dropout (Figure 1(a)). All in all the application of dropout not only mitigates overfitting but also enhances overall model accuracy by promoting more robust feature learning [35].



(a) Standard Neural Network (b) Neural Network with applied dropout

Figure 1: Comparison of a basic Neural Network (a) without applied dropout and (b) with applied dropout. [32]

The following equations describe the forward propagation process in a simple neural network with dropout regularization. Considering a neural network with $L$ hidden layers, where each hidden layer is indexed by $l$, with $l \in \{1, \ldots, L\}$. Let $z^l$ represent the input vector to layer $l$ and $y^l$ denote the output vector, where $y^0 = x$ corresponds to the network input. The weight

matrix and bias vector for layer $l$ are denoted as $w^l$ and $b^l$, respectively. In a standard neural network, the pre-activation value $z_i^{l+1}$ for neuron $i$ in layer $l+1$ is computed as:

$$z_i^{l+1} = y^l w_i^{l+1} + b_i^{l+1}. \tag{1}$$

The activation function $f(\cdot)$ is then applied to produce the output $y_i^{l+1}$:

$$y_i^{l+1} = f\left(z_i^{l+1}\right). \tag{2}$$

When dropout is introduced, a binary mask $r_j^l$ is sampled from a Bernoulli distribution with probability $p$, determining which neurons remain active in layer $l$. The adjusted activations $\tilde{y}^l$ are obtained by element-wise multiplication with $r^l$, leading to a modified pre-activation computation before applying the activation function $f$. The following equations present these principles:

$$\begin{aligned} r_j^l &\sim Bernoulli(p), \\ \tilde{y}^l &= r^l * y^l, \\ z_i^{l+1} &= \tilde{y}^l w_i^{l+1} + b_i^{l+1}. \end{aligned} \tag{3}$$

### 3.2 Monte Carlo Dropout

MCD utilized dropout to approximate the posterior distribution of a neural network [12]. It works by applying dropout not only during training, as traditional dropout does, but also at inference to generate multiple stochastic forward passes through the network. In a standard neural network, dropout randomly deactivates a subset of neurons during training, preventing overfitting. However, in MCD, this stochastic behavior is maintained at inference, effectively turning the network into an ensemble of models with different subsets of active neurons. To estimate the output for a given input, the model performs multiple forward passes $T$, where each forward pass is indexed by $t$, with $t \in \{1, \dots, T\}$, each time using a different dropout mask. Therefore, Equation (3) turns into:

$$z_i^{t,l+1} = \tilde{y}^{t,l} w_i^{l+1} + b_i^{l+1}. \tag{4}$$

This results in a distribution of predictions rather than a single deterministic output. The deterministic prediction can be obtained by averaging these stochastic outputs over $T$ stochastic forward passes to:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} y_L^t, \tag{5}$$

while the variance across them provides an estimate of the underlying model uncertainty:

$$\sigma^2 = \frac{1}{T} \sum_{t=1}^{T} (y_L^t - \hat{y})^2. \tag{6}$$

In case the variance is insufficient, the coefficient of variation (CV) serves as a unitless measure of relative dispersion, enabling comparison between datasets with different means. It is defined as:

$$CV = \frac{S}{\hat{y}}, \tag{7}$$

where $S$ is the standard deviation:

$$S = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(y_L^t - \hat{y})^2}\,. \tag{8}$$

By generating a distribution of predictions instead of a single deterministic output, MCD improves generalization and provides a measure of predictive confidence. This technique serves as a practical approximation of Bayesian inference in deep learning models.

## 4 PROPOSED INPUT PARAMETER SPECIFIC MODEL UNCERTAINTY QUANTIFICATION USING MONTE CARLO DROPOUT

### 4.1 Shortcomings of traditional Monte Carlo dropout

MCD enables the estimation of model prediction uncertainty by maintaining dropout activation during inference, effectively transforming the network into an ensemble of stochastic subnetworks. Therefore, in practice, every neural network layer incorporating dropout contributes to the overall uncertainty estimation, as the variability of each individual layer propagates through the network. By aggregating the uncertainty from all affected layers, MCD provides a measure for the prediction uncertainty of the overall model.

However, when applying MCD in the conventional manner, only the overall model uncertainty can be quantified. A differentiated attribution of uncertainty to individual layers is not feasible, as the method interprets dropout as a stochastic mechanism acting across the entire network. In each forward pass, a random substructure of the network is activated, but the resulting variability in the output is an aggregated effect of all stochastic components. Consequently, MCD does not provide insight into the relative contribution of individual layers to the total predictive uncertainty.

### 4.2 Application of Monte Carlo dropout on individual layers

The proposed application of MCD is aiming to utilize the principles of approximate Bayesian inference individually on single layers of the neural network. The idea is as follows: when applying the MCD principle to a single layer, the resulting output distribution reflects the uncertainty specifically to that one layer. Applying this concept to explicitly embedding or input channels of the neural network enables the UQ to represent the uncertainty of a specific input feature. In model architectures with separate input channels for each individual feature dimension, this method enables an assessment of uncertainty for each individual input parameter. Figure 2(a) shows a flowchart depicting the proposed input parameter-specific Monte Carlo dropout (IPS-MCD). While the overall workflow remains consistent with traditional MCD, the key distinction lies in the targeted application of dropout to a pre-selected input channel during inference.

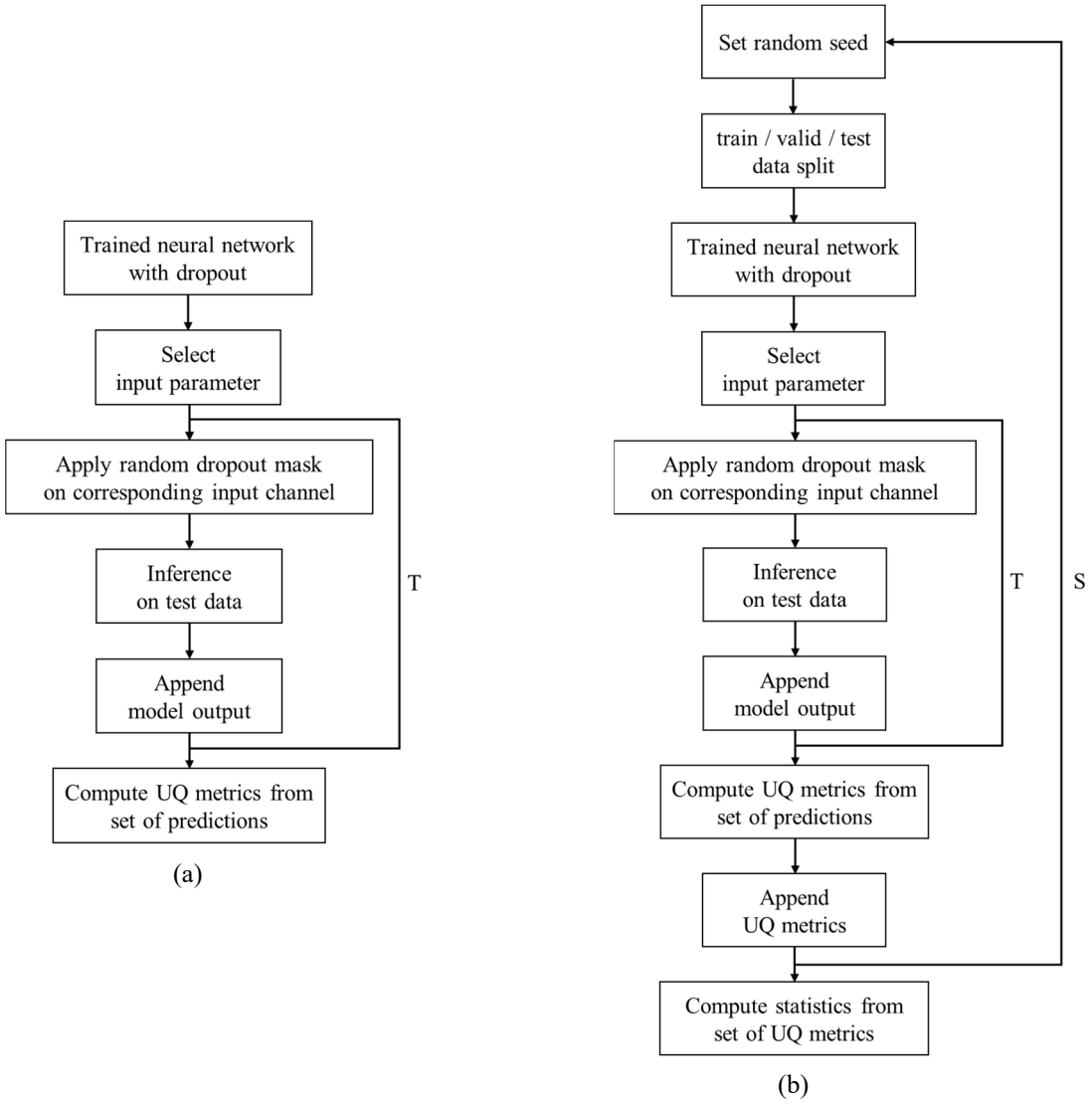



Figure 2: Flowchart of proposed input parameter-specific Monte Carlo dropout, as (a) the main loop and aus (b) the extended multi-seed loop.

## 5 EXPERIMENTAL SETUP AND VALIDATION

### 5.1 Machine Setup

EMF is a high-speed, contact-free manufacturing process that utilizes pulsed magnetic fields to plastically deform electrically conductive workpieces. However, the inherent uncertainties of the process complicate monitoring and control, particularly in predicting the exact forming depth of a workpiece from a single impulse. As part of a data-driven process monitoring model, this depth prediction is utilized to identify input parameter-specific uncertainties, enabling more efficient model improvement and data accumulation. The following paragraph briefly describes the inner workings of the EMF process. For the EMF process, a capacitor battery is charged to a defined voltage and then discharged via the tool coil (inductor) so that damped sinusoidal current flows, causing a transient magnetic field and eddy currents in the electrically conductive workpiece in the opposite direction to the inductor current. Interactions between magnetic field

and current cause Lorentz forces. Thus, the workpiece deforms plastically, when the yield stress is reached. Typical EMF pulses last 15 µs to 100 µs and the magnetic pressure increases up to several hundred mega Pascal [14] [36]. The workpiece is accelerated to speeds of up to several hundred meter per second. Strain rates are in the order of 103 s-1 to 104 s-1 [14]. Figure 3 sketches a free sheet metal forming process. Here, a flat coil with spiral coil winding is applied, but potential forming tasks are not limited to rotationally symmetric cases.

The experimental setup consists of a BlueWave PS 100-25 pulse power generator, which operates with a maximum nominal capacitor charging voltage V of 25 kV and a capacitor charging energy E of 100 kJ. The system includes 22 capacitors, each with a capacitance C of 15 µF, arranged into five capacitor banks that can be connected individually. The resulting capacitor charging energy E can be calculated by

$$E = \frac{1}{2} C U^2. \tag{9}$$

Discharge occurs through gas discharge switches when the capacitor charging voltage V is reached. The switches are connected via coaxial cables to a collector, which is then linked to the inductor using two rigid coaxial conductors. The forming process takes place within a drawing ring with a radius $r_d$ of 20 mm, limiting deformation to a diameter $D_d$ of 100 mm while allowing the rest of the workpiece to form freely. The distance between the coil and workpiece g, capacitance C, and capacitor charging voltage V are varied according to the test design.

For experimental process monitoring, a Rogowski probe (CWT 1500/4/1000, Power Electronics Measurement - PEM) is used to measure the actual transient current flow. The forming depth of the workpiece is recorded using a Keyence LK-H157 laser triangulation measuring system, which relies on a dot-shaped laser beam reflected from the workpiece surface to determine its position. To ensure accurate measurements, the evaluation focuses on the center of the workpiece, where the laser beam remains perpendicular to the surface.

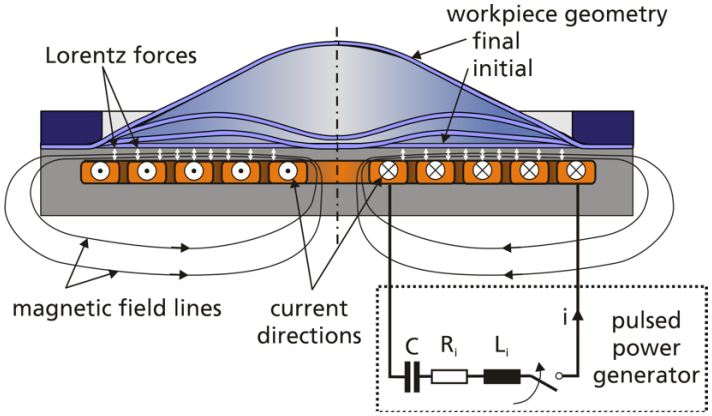



Figure 3: Principle sketch of an electromagnetic sheet metal forming process.

## 5.2 Electromagnetic forming - forming experiments

Forming experiments, with the in Chapter 0 presented EMF machine and monitoring setup, have been conducted. All in all, there are 4 adjustable input parameters, which directly influence the forming depth: the *yield strength* of the used workpiece material, the *distance* between the coil and workpiece and, the *capacitance* and *capacitor charging voltage* of the pulsed power generator. Table **1** presents the value domain of each input parameter resulting in a total of 36 feasible parameter combinations. The two resulting output values are the *current flow*,

measured by the Rogowski probe, and the *forming depth* of the workpiece, measured by the triangulation sensor. The former offers a deeper look into the actual energy transmission from the machine onto the workpiece, while the latter gives a precise numeric value of the true resulting forming of the workpiece. Figure 4 shows a plot of both measured output values for the input parameter combination 44 (yield strength 150 MPa, distance 1.5 mm, capacitance 180 µF and capacitor charging voltage 5.0 kV). The blue graph represents the current flow while the orange graph represents the forming depth over time. To fully grasp the process uncertainty multiple forming experiments are conducted for each parameter combination. In regard to parameter combination 44 the resulting forming depth of 5 different runs are {4.1; 3.66; 3.21; 3.93; 3.45} mm. Figure 5 shows a set of three workpieces after the experiment with different forming depths. The dataset obtained from the forming experiments is not publicly available but can be provided upon individual request.

| **Parameter** | **Values** | **Unit** |
|---|---|---|
| Yield Strength | [125, 150] | MPa |
| Distance | [0.5, 1.5, 2.5] | mm |
| Capacitance | [180, 315] | µF |
| Charging Voltage | [5.0, 7.5, 10.0] | kV |

Table 1: All possible parameter values for the forming experiments.

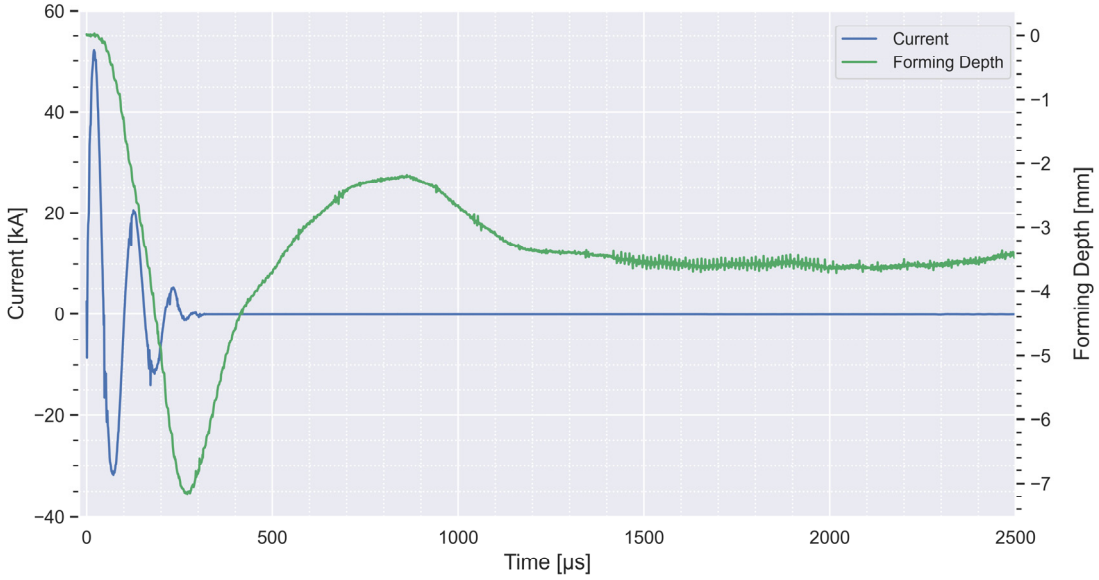



Figure 4: Measured current flow and forming during one electromagnetic impulse.

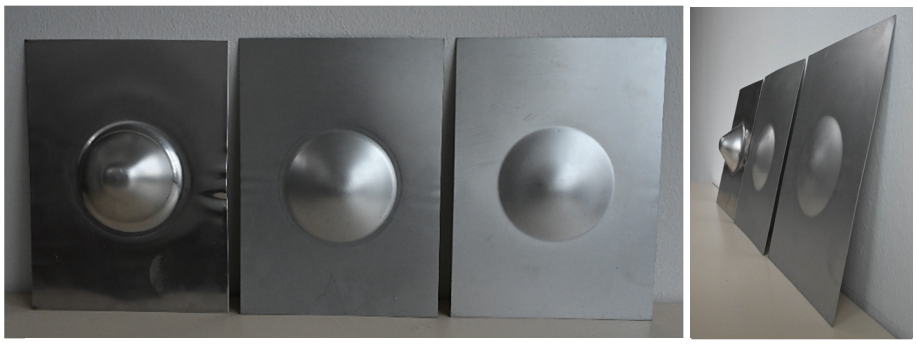


Figure 5: Examples of formed workpieces.

### 5.3 IPS-MCD experimental configuration and hypothesis

To validate the proposed IPS-MCD method, this study incorporates two distinct scenarios with varied experimental configurations, each associated with a specific hypothesis for proof. The model used for the validation is presented in detail in Chapter 5.4.

Scenario 1 – proof of concept

As a proof of concept, the total dataset collected from the initial forming experiments is divided into two equally sized subsets. The key distinction between them lies in the number of samples where the input parameter distance, describing the coil-to-workpiece distance, is set to 1.5 mm. Subset (A) contains significantly fewer samples with this parameter value compared to subset (B). Further details on the exact parameter distribution are provided in Chapter 5.5. The objective of this experimental configuration is to demonstrate that a lower representation of a specific parameter value in subset (A) leads to a higher uncertainty quantification compared to subset (B), thereby validating the effectiveness of the proposed IPS-MCD uncertainty quantification method.

Scenario 2 - strategic design of experiments

A key application of the proposed IPS-MCD method is the derivation from the uncertainty quantification for a strategic DoE in data-sparse environments. Scenario 2 illustrates this use case by applying IPS-MCD to the entire dataset, instead of subsets. The following analysis of the resulting uncertainty metrics (presented in Chapter 5.7 and Chapter 5.8) identifies specific parameter combinations that have the biggest impact on the model's prediction uncertainty. By strategically incorporating additional samples, collected from additional formation experiments, of these critical parameter combinations, the goal is to significantly reduce model uncertainty and improve model accuracy. This approach emphasizes targeted data generation rather than unsystematic or general sampling, thereby maximizing the efficiency and relevance of the collected data.

### 5.4 Model setup

For the verification of the proposed IPS-MCD method, the Temporal Fusion Transformer (TFT) [37] is employed as the base model. Originally developed for time-series forecasting, the TFT offers not only high predictive performance but also structural advantages that align with the characteristics of the given dataset and the implementation of IPS-MCD. A key feature of the TFT is its ability to incorporate static input parameters, allowing the integration of scalar machine and process parameters alongside time-dependent variables. Each static and dynamic input parameter is embedded into a separate input channel before being concatenated and processed by a variable selection network. This specific architectural feature of neural networks enables the input parameter-specific application of MCD. To adapt the TFT for the proposed approach, two modifications are introduced. First, the original multi-horizon forecasting capability of the TFT is restructured into a regression task [2], as the primary objective is the prediction of the formation depths. Second, the input channel architecture of the TFT is modified to enable the application of dropout during inference on pre-selected input channels, thereby allowing uncertainty estimation for specific input parameters.

The model generates quantile predictions and is thus trained using quantile loss as the objective function. The corresponding applied quantile values were $q \in [0.1, 0.5, 0.9]$, where the deterministic prediction is equivalent to $q_{0.5}$.

The hyperparameters of the TFT, specifically state size, number of attention heads, number of Long short-term memory layers, learning rate as well as the dropout probability, were

optimized using Optuna [38]. The optimized hyperparameters obtained were consistently employed throughout the study.

All in all, the input parameters for the TFT are the yield strength, distance, capacitance, charging voltage as static input parameters and the measured current as a variable input parameter. The output of the TFT is the corresponding formation depth obtained from the formation experiments. However, only the four static parameters are used for the proposed IPS-MCD.

### 5.5 Data setup

Scenario 1 – proof of concept

The primary challenge when splitting one dataset into two equally sized subsets with proportional parametric features, except for a specific nested parameter, leads to certain trade-offs. The goal of this proof of concept is to show a difference in the uncertainty quantification in regards to a specific input parameter value imbalance between two comparably equal datasets. Therefore, the complete dataset of 210 samples is divided into two subsets, each containing a total of 174 samples. Per random seed $s$ each subset comprises a core set of shared samples, common to both subsets, along with a distinct set of unique samples specific to each subset. The unique samples within each subset primarily differ in the number of instances where the input parameter distance is set to 1.5 mm. Apart from this distinction, the parameter distributions for yield strength, capacitance, and charging voltage remain approximately equivalent across both subsets, subject to variations introduced by the random seed $s$. Merely, subset (A) contains fewer samples with the distance $\in \{1.5\}$ compared to subset (B). To maintain an equal total sample size across both subsets for training, validation, and testing, the additional samples with distance $\in \{1.5\}$ in subset (B) are counterbalanced by reducing equally the number of samples with distance $\in \{0.5, 2.5\}$. Consequently, subset (B) contains 19 additional samples with distance $\in \{1.5\}$ in the training set, 2 additional samples in the validation set, and 3 additional samples in the test set, with exact values subject to variations introduced by the random seed $s$. As a result, both subsets consist of 139 samples in the training set, 18 samples in the validation set, and 17 samples in the test set.

Scenario 2 – strategic design of experiments

For the second scenario all samples with a total of 210 are used as a base dataset and divided into 168 samples in the training set, 23 samples in the validation set, and 19 samples in the test set. After an initial uncertainty quantification by IPS-MCD is finalized, 29 more samples of overall three specific parameter combinations with the highest CV values are created via additional formation experiments as a strategically extended dataset. These 29 additional samples are split with a 75/25 ration onto the training and validation sets from the base set, not adding anything to the test sets. Resulting in a split of 189 samples in the training set, 31 samples in the validation set, and 19 samples in the test set. Then the uncertainty quantification via IPS-MCD is repeated.

### 5.6 Training setup

The hyperparameter optimization identified an optimal dropout value of 0.15. However, since its effectiveness in achieving the most accurate uncertainty quantification within the proposed IPS-MCD framework is not conclusively determined, the complete training process was repeated multiple times using dropout values from the set of values $D_{Input\ Channel} = \{0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$. These modified dropout values

were exclusively applied to the input channel layers of the associated input parameters to assess their impact on model uncertainty estimation.

Scenario 1 – proof of concept

The training process was designed to ensure robustness as well as comparability in the outcomes. For each subset a total of 70 different random seeds were used to introduce variability in the data splits. Figure 2(b) presents a flowchart of the extended IPS-MCD with a multi random seed approach, where $S$ represents the total number of random seeds used. For each seed $s$ 10 full separate training loops of 1500 epochs each for subset (A) and subset (B) respectively were performed to explore diverse model initialization and data split states. Of these 10 training loops of each subset, only the best models were selected for inference where the minimal validation loss also is identical between subset (A) and subset (B). This guarantees that both models of each subset exhibit equal performance based on validation loss, thus ensuring a certain level of comparability. Additionally, early stopping, was implemented as a regularization technique to mitigate overfitting, where training was halted when the delta between training and validation loss exceeded a predefined threshold. This structured training setup ensured that model comparisons remained consistent, minimizing training variability and enabling a reliable assessment of parameter-specific uncertainties.

Scenario 2 – strategic design of experiments

The same training process described in scenario 1 is applied to scenario 2 with the sole difference being that as there are no data subsets only one model is trained for each of the 70 unique random seeds $s \in S$.

### 5.7 Evaluation metrics and inference setup

The evaluation metrics used for the uncertainty quantification and proof of the proposed IPS-MCD are twofold: The first and main metric is the CV (see Equation (7)) on the outcome distribution. It is derived from the test data, which undergoes $T = 30.000$ forward passes during inference, each using a different dropout mask during the MCD framework. This process introduces variations in the model predictions, generating an outcome distribution from 30.000 predictions for each individual test sample. This distribution is used to compute the CV. Unlike variance, which provides an absolute measure of spread, CV normalizes the standard deviation relative to the mean, allowing for a more scale-independent comparison across different formation depths. This procedure is repeated for each unique random seed $s$, resulting in a set of CV values that capture the variability in model predictions across different training initializations and data splits. This total procedure is performed separately on each input parameter and input channel. Since for every seed $s$ the same model, data splits, and test samples are used for every analyzed input channel during the IPS-MCD, the results remain directly comparable, allowing for a consistent evaluation of the impact of individual parameters on model uncertainty.

Scenario 1 – proof of concept

To achieve uncertainty quantification for each specific input parameter and corresponding input parameter value (see Table 1), the process of performing $T$ forward passes during inference is applied separately to each associated input channel. Consequently, the presence of four distinct input channels results in four independent IPS-MCD computations, enabling the uncertainty contributions associated with each input parameter.

Scenario 2 – strategic design of experiments

To achieve uncertainty quantification for each specific parameter combination, MCD is applied simultaneously to all four input channels, rather than individually. The resulting evaluation metrics are derived from the output distribution and subsequently assigned to each specific parameter combination. After conducting the process with a total of $S$ different random seeds, a comprehensive evaluation of the underlying uncertainty for each parameter combination is obtained.

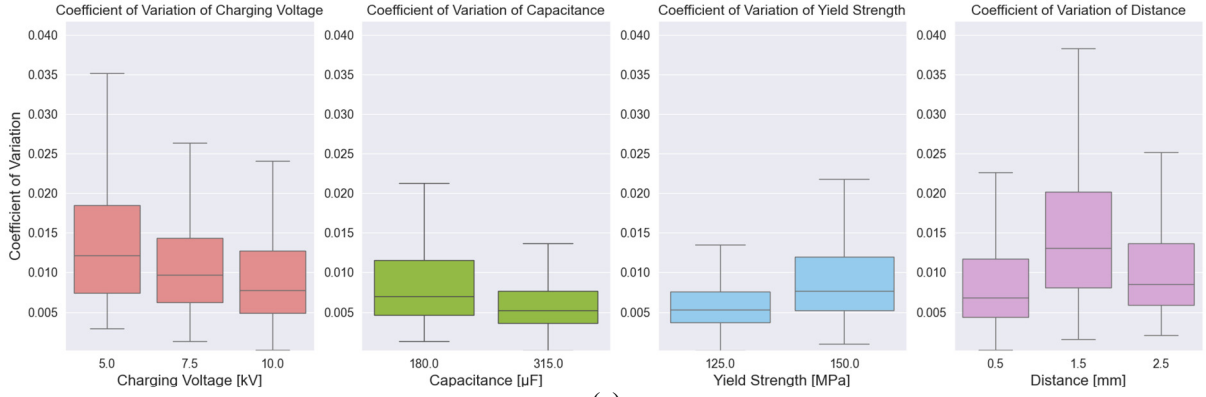
### 5.8 Results

Scenario 1 – proof of concept

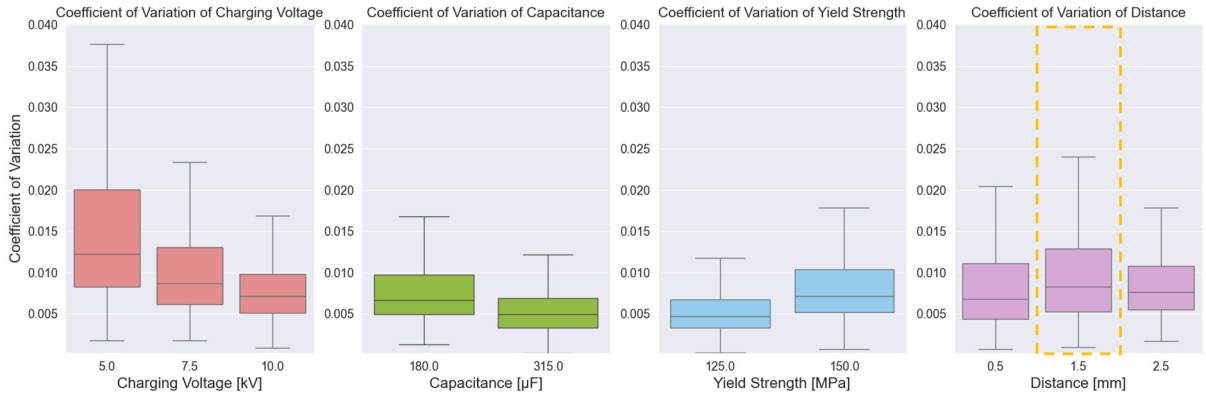
Figure 6 presents the primary results of the IPS-MCD for scenario 1, illustrating the uncertainty quantification of each input parameter value. The results are depicted as boxplots representing the distribution of collected CVs across all seeds for subset (A) in Figure 6(a) and subset (B) in Figure 6(b), with an input channel dropout value of 0.4. The first notable observation is a general conformity in the boxplots across all input parameters, except for the nested difference where distance equals to a value of 1.5 mm between subset (A) and subset (B) (see purple boxplot). As subset (B) contained a significantly higher number of samples where distance was equal to 1.5 mm, this disparity in parameter-specific sample representation (while maintaining equal total sample sizes) is reflected in the resulting model uncertainty quantification (see orange dotted box). This demonstrates the sensitivity of the proposed IPS-MCD framework to variations in input parameter distributions. Table 2 shows the results across all evaluated input channel dropout values. It contains specifically the median as well upper whisker values of the boxplots for the parameter distance with a value of 1.5 mm as well as the resulting percentage reduction. These results are consistent with those shown in Figure 6. The uncertainty represented as a boxplot of all collected CVs across all seeds is for the parameter distance with a value of 1.5 mm in subset (B) visibly lower, with an average reduction of 30%. Table 2 primarily emphasizes the nested differences between subset (A) and subset (B), as including boxplot values for all other input parameters exceeds the scope of this analysis. However, the results presented in Figure 6 for the dropout value of 0.4 are representative for the trends observed across the other dropout values analyzed in this study.

Scenario 2 – strategic design of experiments

Figure 7 displays the primary results of the IPS-MCD for scenario 2, providing the uncertainty quantification of each individual input parameter combination based on a dropout value of 0.4. The vertical axis corresponds to parameter pairs of yield strength and capacitance, whereas the horizontal axis represents parameter pairs of charging voltage and distance. Figure 7(a) specifically shows the results obtained from the base dataset. It is evident that relatively high CV values occur for the parameter combination comprising a yield strength of 150 MPa, capacitance of 180 µF, charging voltage of 5.0 kV, and distances equal to {0.5, 1.5, 2.5} mm, respectively. To explicitly address these comparatively high uncertainty values, an additional 29 samples were strategically incorporated into the base dataset, resulting in a strategically extended dataset as described in Chapter 5.5. These supplementary data include 10 samples for the previously identified parameter combination at a distance of 0.5 mm, 10 samples at 1.5 mm, and 9 samples at 2.5 mm. Figure 7(b) illustrates the results derived from this strategically augmented dataset. It is evident that the CV for these three parameter combinations is noticeably reduced. Meanwhile, the CV values for all remaining parameter combinations remain largely unchanged. Table 3 presents a deeper look into the improvements of the CV between the base and extended dataset on the key parameter combinations using other dropout values, and confirms the results.




(a)

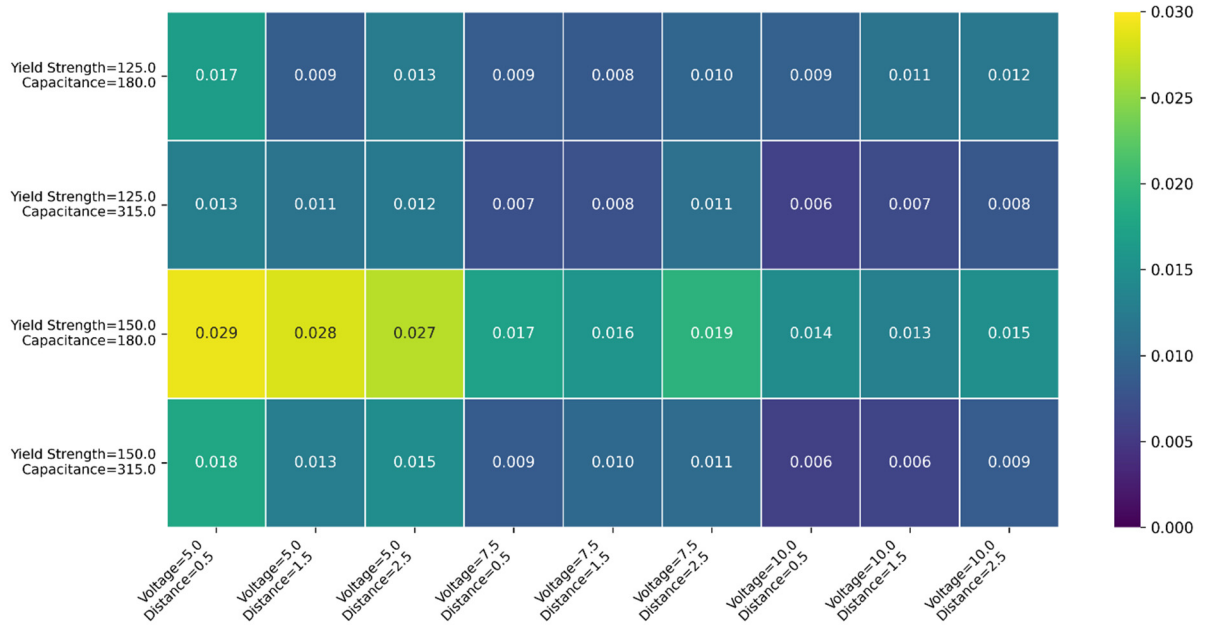



(b)

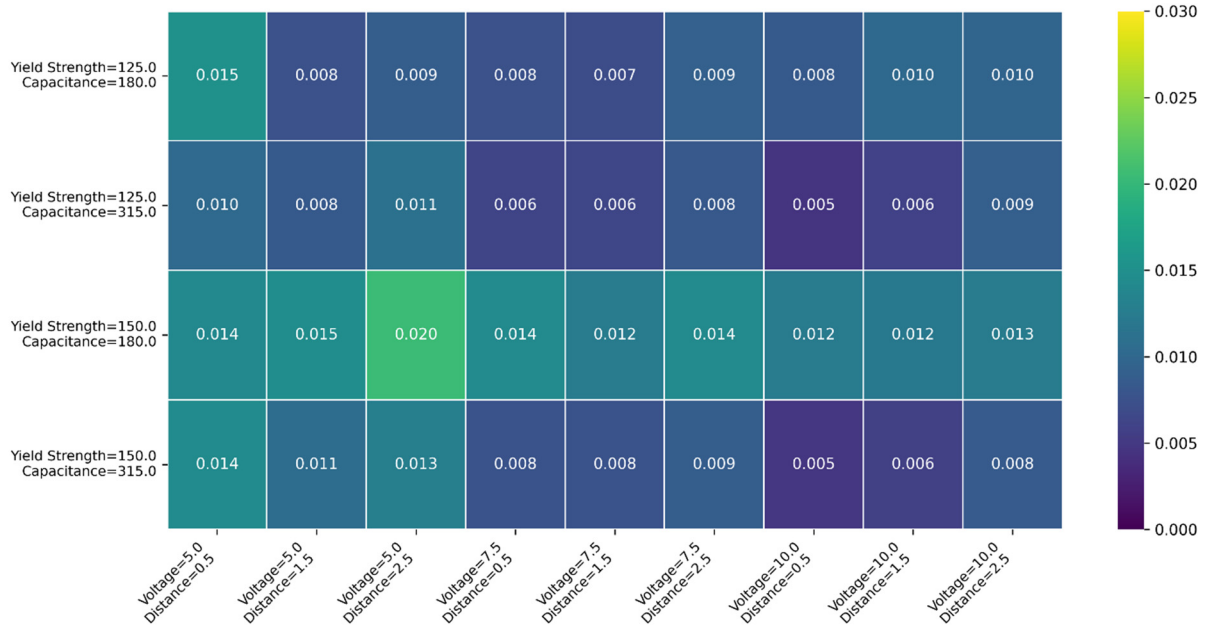
Figure 6: Coefficient of variation of each individual input parameter value using a dropout value of 0.4 for (a) subset (A) and (b) subset (B).

| Dropout Value | Subset (A) [CV] | | Subset (B) [CV] | | Difference [%] |
|---|---|---|---|---|---|
| 0.15 | median | 0.00791 | median | 0.00571 | 27.78 |
| | upper whisker | 0.02048 | upper whisker | 0.01555 | 24.04 |
| 0.20 | median | 0.00929 | median | 0.00601 | 35.26 |
| | upper whisker | 0.02632 | upper whisker | 0.01762 | 33.06 |
| 0.25 | median | 0.00965 | median | 0.00681 | 29.38 |
| | upper whisker | 0.02728 | upper whisker | 0.01915 | 29.80 |
| 0.30 | median | 0.00993 | median | 0.00748 | 24.63 |
| | upper whisker | 0.02842 | upper whisker | 0.01935 | 31.87 |
| 0.35 | median | 0.01147 | median | 0.00702 | 38.79 |
| | upper whisker | 0.02795 | upper whisker | 0.01994 | 28.63 |
| 0.40 | median | 0.01308 | median | 0.00826 | 36.83 |
| | upper whisker | 0.03826 | upper whisker | 0.02398 | 37.30 |
| 0.45 | median | 0.01452 | median | 0.00923 | 36.42 |
| | upper whisker | 0.04220 | upper whisker | 0.02599 | 38.39 |
| 0.50 | median | 0.01555 | median | 0.00962 | 38.09 |
| | upper whisker | 0.04491 | upper whisker | 0.02713 | 39.57 |

Table 2: Boxplot results where distance equals 1.5 mm between subset (A) and subset (B) along different dropout values.




(a)




(b)

Figure 7: Coefficient of variation heat map for every possible input parameter combination for the (a) base dataset and (b) strategically extended dataset

| Dropout Value | Distance Parameter [mm] | Base Dataset [CV] | Strategically Extended Dataset [CV] | Difference [%] |
|---|---|---|---|---|
| 0.15 | 0.5 | 0.021 | 0.007 | 66.67 |
| | 1.5 | 0.021 | 0.010 | 52.38 |
| | 2.5 | 0.015 | 0.012 | 20.00 |
| 0.20 | 0.5 | 0.022 | 0.008 | 63.64 |
| | 1.5 | 0.022 | 0.009 | 59.09 |
| | 2.5 | 0.018 | 0.011 | 38.89 |
| 0.25 | 0.5 | 0.023 | 0.011 | 52.17 |
| | 1.5 | 0.025 | 0.012 | 52.00 |
| | 2.5 | 0.021 | 0.012 | 42.86 |
| 0.30 | 0.5 | 0.025 | 0.011 | 56.00 |
| | 1.5 | 0.025 | 0.012 | 52.00 |
| | 2.5 | 0.020 | 0.016 | 20.00 |
| 0.35 | 0.5 | 0.026 | 0.014 | 46.15 |
| | 1.5 | 0.027 | 0.013 | 51.85 |
| | 2.5 | 0.026 | 0.017 | 34.62 |
| 0.40 | 0.5 | 0.029 | 0.014 | 51.72 |
| | 1.5 | 0.028 | 0.015 | 46.43 |
| | 2.5 | 0.027 | 0.020 | 25.93 |
| 0.45 | 0.5 | 0.029 | 0.018 | 37.93 |
| | 1.5 | 0.033 | 0.018 | 45.45 |
| | 2.5 | 0.032 | 0.022 | 31.25 |
| 0.50 | 0.5 | 0.035 | 0.019 | 45.71 |
| | 1.5 | 0.031 | 0.021 | 32.26 |
| | 2.5 | 0.034 | 0.027 | 20.59 |

Table 3: Comparison of coefficient of variation between base dataset and strategically extended dataset for the parameter combination {YS: 150 MPa, CA: 180 µF, V: 5.0 kV, D: (0.5, 1.5, 2.5) mm}

## 6 DISCUSSION

The experimental results presented in Chapter 5 demonstrate that the targeted application of Monte Carlo dropout exclusively on parameter-specific input channels within a deep learning network can effectively identify the contribution of individual input parameters to the overall model uncertainty. This approach can be utilized on specific parameter values, as presented in Figure 6, as well as parameter combinations, depicted in Figure 7. The resulting uncertainty quantification, which is represented by the coefficient of variation, can be leveraged for a strategic design of experiments. This capability is particularly valuable in sparse-data environments, where obtaining additional data can be time-consuming and/or costly.

Besides the presented proof through the targeted reduction in the uncertainty quantification, the results from the IPS-MCD also effectively represent the underlying uncertainty within the EMF process itself. Due to the internal measurement of the capacitor charging voltage within the pulsed power generator, the measured voltage value is subject to noise effects. As the measured voltage approaches the lower boundary of the measurement range (3.3 kV), the impact of noise on measurement accuracy becomes increasingly pronounced. Consequently, the coefficient of variation is generally elevated at lower capacitor charging energies (see Figure 6). Additionally, the capacitors experience inherent degradation, resulting in a

discrepancy between their nominal and actual capacitance values. Since the current through the inductor is the cumulative sum of the individual capacitor currents, slight variations in switching times can significantly affect the transient behavior of the current curve. With lower capacitance (fewer capacitors) the influence of both effects together has a greater relative influence on the inconsistencies of the transferable energy and thus the uncertainty within the resulting forming depth. Therefore, from a purely technological perspective, the parameters *charging voltage* and *capacitance* have per default a notable impact on the process uncertainty. The results of the proposed IPS-MCD are able to reflect this coherence.

## 7 CONCLUSION

This study successfully advances uncertainty quantification techniques in deep learning networks by strategically applying Monte Carlo dropout to solely input-specific channels. The proposed method enables a more detailed look into the underlying sources of uncertainty, especially in sparse-data environments. The resulting insight can then be utilized for a more strategic design of experiment, effectively saving time and financial resources in the generation of more data. By selectively targeting specific input parameters, the proposed method, which is introduced as input parameter-specific Monte Carlo dropout (IPS-MCD), accurately identifies parameter values as well as parameter combinations associated with high uncertainty. Consequently, it is able to quantify uncertainty in critical areas and enables a targeted reduction in uncertainty, thereby enhancing model reliability and predictive confidence.

## 8 ACKNOWLEDGEMENTS

The authors gratefully acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - funding number IH 123/32-1.

## REFERENCES

[1] E. Boos, M. Schwarzenberger, M. Jaretzki, and S. Ihlenfeldt, “Melt pool monitoring using fuzzy based anomaly detection in laser beam melting,” in *Proceedings of the Metal Additive Manufacturing Conference, Örebro, Sweden*, 2020, pp. 1–11.

[2] E. Boos, J. Zimmermann, H. Wiemer, and S. Ihlenfeldt, “Investigation of alternative attention modules in transformer models for remaining useful life predictions: addressing challenges in high-frequency time-series data,” *Procedia CIRP*, vol. 122, pp. 85–90, 2024, doi: 10.1016/j.procir.2024.01.012.

[3] C. Li, Y. Chen, and Y. Shang, “A review of industrial big data for decision making in intelligent manufacturing,” *Engineering Science and Technology, an International Journal*, vol. 29, p. 101021, May 2022, doi: 10.1016/j.jestch.2021.06.001.

[4] A. Schimke, O. Gnepper, and E. Boos, “Data-Driven Damage Assessment of Axial Piston Pumps,” in *2024 IEEE International Conference on Engineering, Technology, and Innovation (ICE/ITMC)*, Funchal, Portugal: IEEE, Jun. 2024, pp. 1–9. doi: 10.1109/ICE/ITMC61926.2024.10794220.

[5] X. Bai *et al.*, “Explainable deep learning for efficient and robust pattern recognition: A survey of recent developments,” *Pattern Recognition*, vol. 120, p. 108102, Dec. 2021, doi: 10.1016/j.patcog.2021.108102.

[6] M. Abdar *et al.*, “A review of uncertainty quantification in deep learning: Techniques, applications and challenges,” *Information Fusion*, vol. 76, pp. 243–297, Dec. 2021, doi: 10.1016/j.inffus.2021.05.008.

[7] H. Li, J. Jiao, Z. Liu, J. Lin, T. Zhang, and H. Liu, “Trustworthy Bayesian deep learning framework for uncertainty quantification and confidence calibration: Application in machinery fault diagnosis,” *Reliability Engineering & System Safety*, vol. 255, p. 110657, Mar. 2025, doi: 10.1016/j.ress.2024.110657.
[8] A. Kendall and Y. Gal, “What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision?”.
[9] E. Hüllermeier and W. Waegeman, “Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods,” *Mach Learn*, vol. 110, no. 3, pp. 457–506, Mar. 2021, doi: 10.1007/s10994-021-05946-3.
[10] L. Basora, A. Viens, M. A. Chao, and X. Olive, “A benchmark on uncertainty quantification for deep learning prognostics,” *Reliability Engineering & System Safety*, vol. 253, p. 110513, Jan. 2025, doi: 10.1016/j.ress.2024.110513.
[11] A. Loquercio, M. Segu, and D. Scaramuzza, “A General Framework for Uncertainty Estimation in Deep Learning,” *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3153–3160, Apr. 2020, doi: 10.1109/LRA.2020.2974682.
[12] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning,” in *Proceedings of The 33rd International Conference on Machine Learning*, PMLR, Jun. 2016, pp. 1050–1059. Accessed: Jan. 28, 2025. [Online]. Available: https://proceedings.mlr.press/v48/gal16.html
[13] M. Valdenegro-Toro and D. Saromo, “A Deeper Look into Aleatoric and Epistemic Uncertainty Disentanglement,” Apr. 20, 2022, *arXiv*: arXiv:2204.09308. doi: 10.48550/arXiv.2204.09308.
[14] V. Psyk, D. Risch, B. L. Kinsey, A. E. Tekkaya, and M. Kleiner, “Electromagnetic forming—A review,” *Journal of Materials Processing Technology*, vol. 211, no. 5, pp. 787–829, May 2011, doi: 10.1016/j.jmatprotec.2010.12.012.
[15] J. Gawlikowski *et al.*, “A Survey of Uncertainty in Deep Neural Networks,” Jan. 18, 2022, *arXiv*: arXiv:2107.03342. doi: 10.48550/arXiv.2107.03342.
[16] J. Gao, M. Chen, L. Xiang, and C. Xu, “A Comprehensive Survey on Evidential Deep Learning and Its Applications,” Sep. 07, 2024, *arXiv*: arXiv:2409.04720. doi: 10.48550/arXiv.2409.04720.
[17] A. Malinin and M. Gales, “Predictive Uncertainty Estimation via Prior Networks,” Nov. 29, 2018, *arXiv*: arXiv:1802.10501. doi: 10.48550/arXiv.1802.10501.
[18] J. van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, “Uncertainty Estimation Using a Single Deep Deterministic Neural Network”.
[19] J. Cheung, S. Rangarajan, A. Maddocks, X. Chen, and R. Chandra, “Quantile deep learning models for multi-step ahead time series prediction,” Nov. 24, 2024, *arXiv*: arXiv:2411.15674. doi: 10.48550/arXiv.2411.15674.
[20] D. J. C. MacKay, “Bayesian neural networks and density networks,” *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 354, no. 1, pp. 73–80, Jan. 1995, doi: 10.1016/0168-9002(94)00931-7.
[21] T. Huix, S. Majewski, A. Durmus, E. Moulines, and A. Korba, “Variational Inference of overparameterized Bayesian Neural Networks: a theoretical and empirical study,” Jul. 08, 2022, *arXiv*: arXiv:2207.03859. doi: 10.48550/arXiv.2207.03859.
[22] A. Shamsi, H. Asgharnezhad, A. Tajally, S. Nahavandi, and H. Leung, “An Uncertainty-aware Loss Function for Training Neural Networks with Calibrated Predictions,” Feb. 06, 2023, *arXiv*: arXiv:2110.03260. doi: 10.48550/arXiv.2110.03260.
[23] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles,” Nov. 04, 2017, *arXiv*: arXiv:1612.01474. doi: 10.48550/arXiv.1612.01474.
[24] I. Ezhov *et al.*, “Geometry-Aware Neural Solver for Fast Bayesian Calibration of Brain Tumor Models,” *IEEE Transactions on Medical Imaging*, vol. 41, no. 5, pp. 1269–1278, May 2022, doi: 10.1109/TMI.2021.3136582.
[25] B. C. Haas, D. Kalyani, and M. S. Sigman, “Applying statistical modeling strategies to sparse datasets in synthetic chemistry,” *Sci. Adv.*, vol. 11, no. 1, p. eadt3013, Jan. 2025, doi: 10.1126/sciadv.adt3013.

[26] H. Song *et al.*, “Adaptive Experiments Under High-Dimensional and Data Sparse Settings: Applications for Educational Platforms,” 2025, *arXiv*. doi: 10.48550/ARXIV.2501.03999.
[27] I. Steponavičė, M. Shirazi-Manesh, R. J. Hyndman, K. Smith-Miles, and L. Villanova, “On Sampling Methods for Costly Multi-Objective Black-Box Optimization,” in *Advances in Stochastic and Deterministic Global Optimization*, vol. 107, P. M. Pardalos, A. Zhigljavsky, and J. Žilinskas, Eds., in Springer Optimization and Its Applications, vol. 107. , Cham: Springer International Publishing, 2016, pp. 273–296. doi: 10.1007/978-3-319-29975-4_15.
[28] A. Hoarau, B. Quost, S. Destercke, and W. Waegeman, “Reducing Aleatoric and Epistemic Uncertainty through Multi-modal Data Acquisition,” Jan. 30, 2025, *arXiv*: arXiv:2501.18268. doi: 10.48550/arXiv.2501.18268.
[29] T. Blau, E. V. Bonilla, I. Chades, and A. Dezfouli, “Optimizing Sequential Experimental Design with Deep Reinforcement Learning”.
[30] K. Chaloner and I. Verdinelli, “Bayesian Experimental Design: A Review,” *Statist. Sci.*, vol. 10, no. 3, Aug. 1995, doi: 10.1214/ss/1177009939.
[31] T. Rainforth, A. Foster, D. R. Ivanova, and F. B. Smith, “Modern Bayesian Experimental Design,” Nov. 29, 2023, *arXiv*: arXiv:2302.14545. doi: 10.48550/arXiv.2302.14545.
[32] P. Novello, G. Poëtte, D. Lugato, and P. Congedo, “Leveraging Local Variation in Data: Sampling and Weighting Schemes for Supervised Deep Learning,” *J Mach Learn Model Comput*, vol. 3, no. 1, pp. 77–105, 2022, doi: 10.1615/JMachLearnModelComput.2022041819.
[33] T. Xie *et al.*, “Structural-Entropy-Based Sample Selection for Efficient and Effective Learning,” Oct. 05, 2024, *arXiv*: arXiv:2410.02268. doi: 10.48550/arXiv.2410.02268.
[34] J. Kukačka, V. Golkov, and D. Cremers, “Regularization for Deep Learning: A Taxonomy,” Oct. 29, 2017, *arXiv*: arXiv:1710.10686. doi: 10.48550/arXiv.1710.10686.
[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”.
[36] Z. Lai *et al.*, “A comprehensive electromagnetic forming approach for large sheet metal forming,” *Procedia Engineering*, vol. 207, pp. 54–59, 2017, doi: 10.1016/j.proeng.2017.10.737.
[37] B. Lim, S. O. Arik, N. Loeff, and T. Pfister, “Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting,” *arXiv:1912.09363 [cs, stat]*, Sep. 2020, Accessed: May 04, 2022. [Online]. Available: http://arxiv.org/abs/1912.09363
[38] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” Jul. 25, 2019, *arXiv*: arXiv:1907.10902. Accessed: Nov. 10, 2023. [Online]. Available: http://arxiv.org/abs/1907.10902