Uncertainty-aware Virtual Sensors for Cyber-Physical Systems

Sagar Sen, Erik Johannes Husom, Arda Goknil, Simeon Tverdal, Phu Nguyen
SINTEF, Oslo, Norway

***Abstract*—Virtual sensors in Cyber-Physical Systems (CPS) are AI replicas of physical sensors that can mimic their behavior by processing input data from other sensors monitoring the same system. However, we cannot always trust these replicas due to uncertainty ensuing from changes in environmental conditions, measurement errors, model structure errors, and unknown input data. An awareness of numerical uncertainty in these models can help ignore some predictions and communicate limitations for responsible action. We present a data pipeline to train and deploy *uncertainty-aware virtual sensors* in CPS. Our virtual sensor based on a *Bayesian Neural Network* (BNN) predicts the expected values of a physical sensor and a standard deviation indicating the degree of uncertainty in its predictions. We discuss how this uncertainty awareness bolsters trustworthy AI using a vibration-sensing virtual sensor in automotive manufacturing.**

CYBER-PHYSICAL SYSTEMS (CPS) have been increasingly implanted in recent years with Artificial Intelligence (AI) and, in particular, *Deep Learning Models* (DLM). DLMs are arguably the only viable tool purported to make accurate decisions and predictions from a juggernaut of data streams generated by devices interconnected through the Internet of Things (IoT) for sensing, computation, control, actuation, and networking. Virtual sensors (also called soft sensors) [1] are DLMs that are AI replicas (software artifacts replicating the output of a physical component/sensor of a CPS by learning its correlated behavior with one or more different physical artifacts in the CPS) [2] of potentially billions of physical sensors (e.g., pressure, temperature, humidity, speed, force, vibration, and position sensors) deployed in CPS. They can kick in for degrading physical sensors operating in harsh environments [3]. For instance, a virtual sensor can mimic a grinding force sensor in CPS for manufacturing by using the spindle load measurement as input. Virtual sensors are vulnerable to various forms of *uncertainty* (e.g., measurement error/noise in any of the sensors, distributional shifts w.r.t training data ensuing from variability in environmental conditions, and errors in the model architecture), making these replicas hard to trust. Therefore, our motivation is to answer if and how we can engineer virtual sensors that can quantify the degree of their uncertainty to ensure their trustworthiness in industrial settings.

We present a Machine Learning (ML) pipeline to train and deploy *uncertainty-aware virtual sensors* (i.e., AI replicas of faulty physical sensors with uncertainty estimation of their predictions) in a CPS. A CPS is assumed to contain one or more input sensors that maintain a *latent relationship* with the faulty physical sensor to predict its behavior. Our pipeline extracts statistical

properties (features) from time series data, trains Bayesian Neural Network (BNN) models, and deploys these models as an uncertainty-aware virtual sensor service. Uncertainty awareness refers to the ability of a system to recognize, understand, and respond to uncertainty and variability in its inputs, processes, and outputs. It involves differentiating types of uncertainty, quantifying its degree, and using this information to make decisions and take appropriate actions.

Our virtual sensors process time-varying values from input sensors and return two outputs: the mean prediction of predicted data (i.e., "predicted data" in Figure 1) for the faulty sensor and the standard deviation from the mean prediction. The uncertainty is quantified as a two-tailed prediction interval using the standard deviation of a set of predictions sampled from the BNN model. Standard deviation is a widely used measure of variability or dispersion in a data set. This uncertainty quantification using standard deviation enables human experts to gauge the potential variability in a prediction and take the decision with a *pinch of salt*. It also helps them distinguish epistemic (uncertainty in the model) from aleatoric uncertainty (uncertainty due to randomness in input sensor data). Epistemic uncertainty (also known as *model uncertainty*) arises due to the lack of knowledge or incomplete understanding of the underlying model. Given available input sensor data, we need to verify if variations in the virtual sensor can affect the variability in the output. Aleatoric uncertainty (known as *data uncertainty*) arises due to the inherent randomness or variability in the input data to the virtual sensor. This study focuses on aleatoric uncertainty in physical sensors affecting BNNs due to issues like sensor freezing, shifts, drifts, and precision degradation. We exclude epistemic uncertainty, which can be captured through various methods (e.g., Monte Carlo dropout, variational inference, or Markov Chain Monte Carlo sampling). Aleatoric uncertainty, more prominent in our Industrial Internet of Things (IIoT) data, is quantified as a standard deviation from mean prediction and analyzed as trends in uncertainty. It cannot be reduced through better modeling or training, as it is a fundamental property of the data. Standard deviation can also be used as a performance metric to optimize the training of virtual sensors (given the same input data) in order to reduce aleatoric uncertainty. Aleatoric uncertainty due to physical sensor failures exhibits trends such as a sudden increase in uncertainty due to the freezing of a sensor or a linear increase in uncertainty due to precision degradation. Epistemic uncertainty is an aggregation of the behavior of a BNN with different seeds for Monte Carlo simulation. It is an aggregate range of values around a mean and does not necessarily exhibit identifiable trends.

The uncertainty quantification in Deep Neural Networks (DNN) [4] can also be achieved through test-time augmentation (making variations in input data to quantify uncertainty in output) and ensemble neural networks (training multiple models and aggregating the mean and variability in prediction). Our goal is not to compare different uncertainty quantification approaches but to operationalize trustworthy AI for virtual sensors by using uncertainty estimates.

Trustworthy AI requirements have been proposed by public entities such as NIST[1] and the European commission's high-level expert group on trustworthy AI (AI-HLEG) [5]. AI-HLEG has formulated requirements (e.g., human agency and oversight, technical robustness and safety, diversity, non-discrimination and fairness, and transparency) that should be implemented and assessed throughout an AI system's life cycle ([5], p. 15). We discuss how uncertainty-aware virtual sensors bolster trustworthy AI in CPS based on the AI-HLEG requirements. We support our arguments using a real-world case study for automotive manufacturing where we develop a virtual sensor for a vibration sensor on a machine tool using data from another vibration sensor in close proximity to it.

Virtual Sensors and Uncertainty Estimation in CPS Operations

Virtual sensors [6], [1], [7] and uncertainty-aware virtual sensors [8] are not new in the literature. Kabadayi et al. [6] first provided the abstraction of virtual sensors, i.e., virtual sensors abstract a set of physical sensors and operations performed on them. Our virtual sensors make predictions based on input data from physical

[1]https://www.nist.gov/itl/ai-risk-management-framework

sensors. They can be called secondary sensors (physical sensors are primary ones). We only consider virtual sensors replacing physical sensors, although they can also act as secondary sensors for software probes instrumenting running code and monitoring functionality of software systems.

Some approaches [9], [10], [11], [12], [13], [14] apply DL techniques to develop virtual sensors. None of them provide predictions with a degree of uncertainty. This limitation can lead to severe problems where incorrectly predicted values from virtual sensors using highly noisy signals or used for cases not included in the training data are used for decision-making.

Lee et al. [8] propose uncertainty-aware virtual sensors using Bayesian Recurrent Neural Networks to estimate industrial process variables and quantify the uncertainties associated with each variable. However, they do not provide any pipeline to engineer such virtual sensors. They also do not discuss the deployment of these virtual sensors in industrial settings. On the other hand, we introduce an ML pipeline that enables engineers to set different DL parameters (including the selection of Bayesian models such as Bayesian Convolutional Neural Networks and Recurrent Neural Networks) and combine various statistical features in the input domain while engineering virtual sensors for industrial settings.

Different than the work by Lee et al., our research is primarily necessity-driven/problem-driven since erroneous sensor data is a recurring problem in CPS, e.g., in manufacturing. Novel results in ML/DL with uncertainty estimation using Bayesian approaches are very timely for the erroneous data problem in CPS. We present a framework grounded on uncertainty-aware virtual sensors to solve prediction problems based on sensor data in CPS. Furthermore, we discuss how uncertainty estimation enhances the trustworthiness of virtual sensors for stakeholders.

Our Approach for Uncertainty-aware Virtual Sensors

A CPS employs several sensors to monitor a physical process simultaneously and take control of actions. If any of these sensors fails or becomes faulty, it is often possible to estimate its correct values based on data acquired simultaneously in other sensors. An *ML model* can be used to learn latent relationships between time-varying data from input sensors to predict time-varying measurements of a faulty/failed sensor. It can be deployed as a service (with an API), which essentially becomes a replica (virtual sensor) of the faulty sensor. The input and output datasets for training should ideally be complete, accurate, timely, valid, and without faults. However, in reality, training data (input/output pairs) is often missing for rare events such as sensor faults, environmental noise, and outliers. Therefore, the information of uncertainty degree must accompany predictions for us to trust them.

Our ML pipeline in Figure 1 creates uncertainty-aware virtual sensors that output both the mean predicted value of the target sensor and a prediction interval. This interval can be regarded as how confident we are in a prediction. It, if small, indicates that the ML model predicts with high certainty as patterns in the input are most likely close to patterns already seen in training. If the interval is large, we are less confident about the prediction as the model does not recognize the patterns in the input (e.g., unforeseen during training, including environmental noise and rare events). The pipeline trains BNNs [15] that treat weights in neural networks as marginal distributions that best fit the data. Since the weights of a BNN are probability distributions, we can produce multiple predictions for the same input by sampling from these distributions. The final predicted value will be the mean of these predictions, while the uncertainty is represented by a two-tailed interval of $\pm$ the standard deviation of the predictions. It helps ascertain the degree of prediction uncertainty and, consequently, helps address high-level concepts of trustworthy AI.

Pipeline Stages

The pipeline *input* is multi-variate time series data X from input sensors and univariate time series data y from a target sensor (Figure 1). The *output* is an uncertainty-aware virtual sensor that predicts the mean value of target sensor $\bar{y}$ and standard deviation σ representing the prediction interval given production input data. The input (X, y) goes through three stages:

Stage 1: Data pre-processing. This stage prepares input data for BNN in several sub-stages: *data profiling*, *data cleaning*, *feature engineer-*

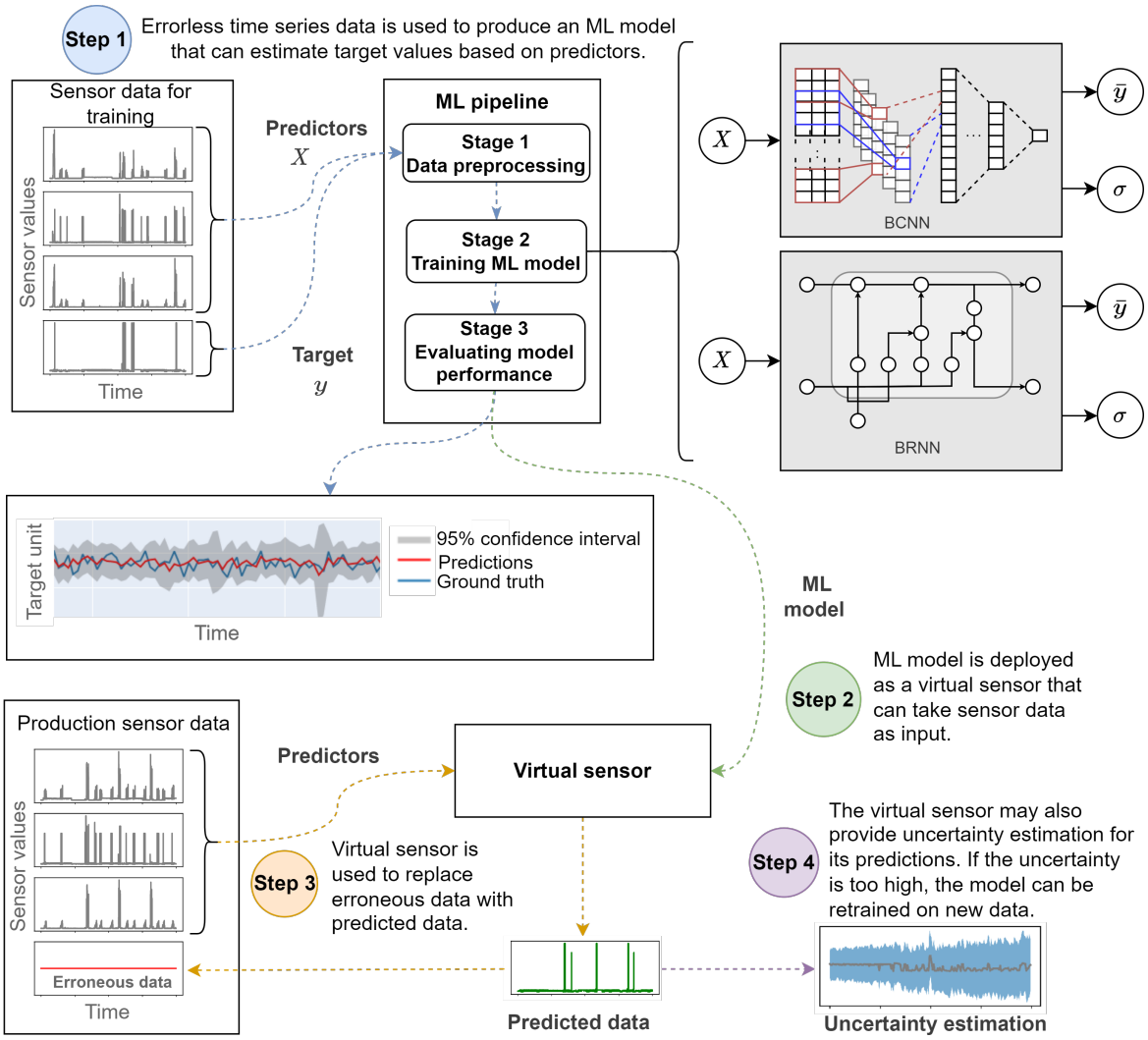

Figure 1. Machine learning pipeline using Bayesian Neural Networks (BNNs) to produce models capable of uncertainty estimation.

ing, *splitting test/training data*, *data scaling*, and *splitting data into sub-sequences*. Data profiling computes the non-linear *maximum information coefficient* and linear *Pearson's correlation coefficient* to find correlations between data columns of different sensors. It generates statistical quantities for each column and alerts if any column has several zeros or missing values. Data cleaning uses this output to remove unwanted data (e.g., columns with several constant or null values).

Feature engineering extracts statistical properties (*features*) from raw input data that exhibit invariance to noise. Furthermore, the feature-based representations of time-series data [16] perform well in classifying tasks at a fraction of the computational cost of processing raw time-series. The input and output data are split into *training* and *test* datasets. The training set is used to train the ML model and tune hyper-parameters in Stage 2. The test dataset is locked away during training and used as an unbiased dataset to evaluate virtual sensor performance in Stage 3.

The datasets contain measurements from different sensors with varying value ranges and are, thus, scaled [17] for a comparable influence during training. The training (including validation dataset) and test datasets are *restructured* into input and output sub-sequences of the specified *window size* since predictions rely on a window of time-varying observations from input sensors

and the desired window of output values.

Stage 2: Training ML model. Our pipeline uses the input and output sub-sequences (of chosen input and output *window sizes*) from Stage 1 to train the ML model. It enables the specification of learning parameters and the selection of ML model types (architectures). We consider Bayesian Convolutional Neural Networks (BCNN) and Bayesian Recurrent Neural Networks (BRNN) to predict the time-varying values of a virtual sensor. Before training, the pipeline sets apart a small portion of the training data (e.g., 20%) to use as a *validation dataset*. It automatically stops training the model if the prediction error of the validation dataset stops improving, preventing the over-fitting of the model to the training data. It saves the model for evaluation.

Stage 3: Evaluating model performance. The test dataset is an unforeseen dataset used to evaluate the model performance to minimize bias due to hyper-parameter tuning in Stage 2. We compare the model output and the ground truth to assess how well the model predicts the target variable. To this end, the pipeline generates the plots of predictions on test data. We use Mean Squared Error (MSE), coefficient of determination (R^2 score), and Mean Absolute Percentage Error (MAPE) to evaluate the model performance.

Step 1 in Figure 1 involves using multi-variate input sensor data X to train an ML model to predict the values of target sensor y. This model is created using either a BCNN or a BRNN. While traditional neural networks output a single value prediction, BNNs produce probability distribution for each prediction, represented by mean value $\bar{y}$ and standard deviation σ. The resulting ML model is deployed as a virtual sensor that can take sensor data from production as input (Step 2). It replaces erroneous data from the target sensor (Step 3). The inference procedure of a BNN is more computationally expensive than traditional neural networks since it involves drawing several samples from the probability distributions of the model's parameters and computing multiple possible outputs. Therefore, BNNs will place higher requirements on the computational hardware if deployed for time-sensitive high-frequency data. The virtual sensor is operational along with the output physical sensor increasing redundancy in the CPS. It can provide values for the physical sensor if it has erroneous behavior outside a pre-defined amplitude and frequency range due to problems such as high SNR, aliasing, and jitter. However, the virtual sensor relies on input sensors which can experience data drift due to sensor faults or changes in a process governed by the CPS. Therefore, we estimate uncertainty in predictions by the virtual sensor given by the standard deviation σ for each prediction (Step 4). Higher uncertainty might be a symptom of drift and necessitate updating the model in the virtual sensor or replacing the input sensors.

We can apply our pipeline to non-time series input data (e.g., tabular data) by using single data points instead of sub-sequences (in practice, using a sub-sequence with a window size of 1). Moreover, one should use fully-connected BNNs instead of recurrent and convolutional networks since the latter networks are applied to sequences of data points.

If a physical sensor input to a virtual one fails, the virtual one cannot make predictions. When any input is missing, the virtual sensor is ineffective. We can use the virtual sensor as input to another virtual sensor. In that case, the uncertainty in the output of the first virtual sensor propagates to the second one. We only focus on estimating uncertainty in a virtual sensor taking input from physical sensors.

Deployment of Virtual Sensors

A virtual sensor embodies the trained, validated, and evaluated ML model as a service with an API. The API is invoked using sub-sequences of data from input sensors and returns a set of target sensor values with time stamps. Since the pipeline trains the ML model on input data features extracted from raw data and bounded by a scaling operation, the model cannot always use the raw input sequences as they are. The feature engineering operations require using ML libraries such as Scikit-learn and TensorFlow. Therefore, parts of the pipeline used in inference, such as code to compute engineered features (Stage 1), scaler (Stage 1), and the model (Stage 2) with all its dependencies (e.g., ML libraries), are packaged as a standalone container (e.g., docker).

Uncertainty Estimation in Virtual Sensors

Uncertainty in the virtual sensor predictions can be classified as (i) *aleatoric uncertainty* (uncertainty in the model output) and (ii) *epistemic uncertainty* (uncertainty in the model's weights).

Aleatoric uncertainty is high when the ML model tries to infer from out-of-distribution input data (not foreseen during training) from sensors experiencing precision degradation, freezing, and environmental noise (e.g., electromagnetic coupling with a nearby power line). It can further be classified as (a) *heteroscedastic aleatoric uncertainty*, where each observation (x, y) has a different noise extent, and (b) *homoscedastic aleatoric uncertainty*, where the noise level of the observations is identical.

Uncertainty in the model weights (epistemic) may occur due to randomness in training. For instance, BNNs are stochastic neural networks that employ methods involving randomization (Markov Chain Monte Carlo (MCMC) [18] sampling probability distributions for sophisticated integrals). Different random seeds used in MCMC lead to different weights during training, causing uncertainty in the model output.

The standard deviation σ generated by an uncertainty-aware virtual sensor is the sum of aleatoric and epistemic uncertainty. The prediction interval is a combination of both forms of uncertainty. Testing the virtual sensor by inducing faults in input sensors is one way to see the effect of *out-of-distribution* inputs (e.g., white noise, freezing, precision degradation) on prediction intervals. If the intervals are higher than usual, the virtual sensor is able to detect the fault and differentiate between aleatoric and epistemic uncertainty. We can determine epistemic uncertainty by training the BNN with different seed values for random number generation. The resulting prediction interval due to the different models while keeping input data unchanged is the epistemic uncertainty of the BNN. The average variability of the output coming from the model is anti-proportional to the validation accuracy. The differentiation of epistemic uncertainty from aleatoric uncertainty is an active research area [19] beyond the scope of this article.

We use uncertainty-aware virtual sensors to determine aleatoric uncertainty due to inputs affected by faults in input sensors. The prediction interval estimated by a virtual sensor serves as information to address the trustworthiness of ML models used as replicas of CPS sensors.

Case Study: Virtual Sensor for Vibration Prediction in Manufacturing

We applied our pipeline to a case study where accelerometers for vibration sensing were installed on a Computerized Numerical Control (CNC) grinding machine. The machine is used to manufacture bearings for the automotive industry. We acquired vibration data from the machining of bearings (more specifically, grinding of the inner ring) using accelerometers recording data at 32768 Hz.

Accelerometers may experience faults (e.g., freezing and precision degradation) due to environmental factors (e.g., electromagnetic interference [3]). We created an uncertainty-aware virtual sensor to replace one faulty accelerometer when data from another accelerometer is simultaneously available. We considered two high-frequency accelerometers used to measure vibration in the vertical and horizontal directions of the wheel head spindle (see Figure 2(a)). The virtual sensor was based on a BCNN and predicted *wheel spindle vertical acceleration* by using input data from *wheel spindle horizontal acceleration*. It obtained an R^2 score of 0.9 on unforeseen test data. Such a high score was expected as the grinding process for bearings is very repetitive. The spike observed around timestep 300 in Figure 2 represents a dramatic increase in uncertainty in the prediction. Likely, the behavior displayed by the sensor in that instance was not present in the training dataset, which would cause the increase due to the unforeseen behavior. It is also possible that the spike represents an error or fault on behalf of the machining process or sensor reading.

We illustrate the performance of uncertainty-aware virtual sensors with two sensor fault types: *freezing* and *precision degradation*. Uncertainty should increase when there are faults in the input sensor (the sensor monitoring the wheel spindle horizontal acceleration in our case). Figure 2 shows the prediction of wheel spindle vertical acceleration with uncertainty for freezing and precision degradation. It zooms into 0.04 seconds (1600 data points) of the *prediction value* (the mean predicted value by the BNN) and the

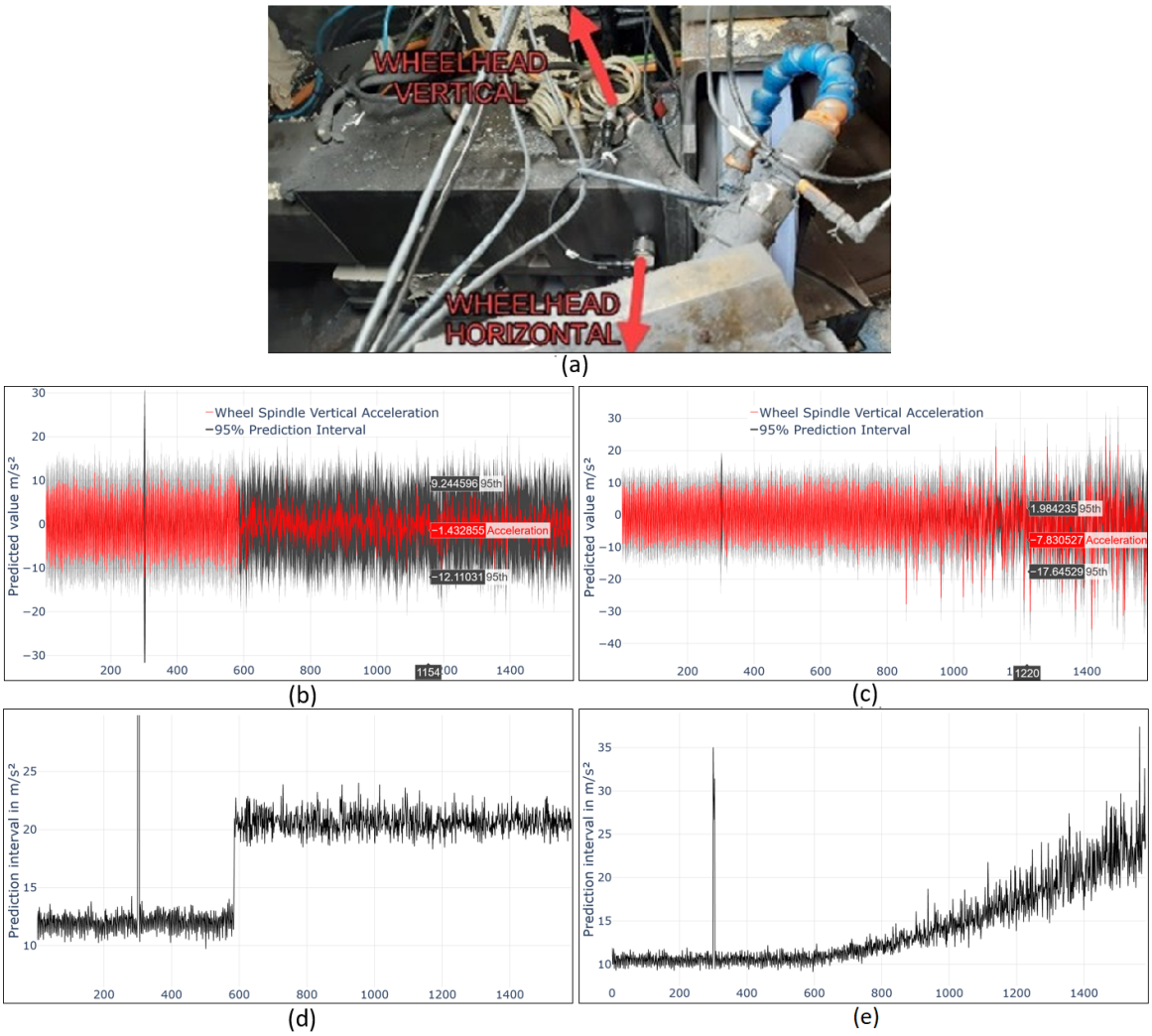

Figure 2. (a) High-frequency accelerometers installed on vertical and horizontal axes of the grinding wheel spindle head to measure vibration. Each arrow indicates the axis in which the corresponding accelerometer measures the acceleration. (b) Uncertainty in virtual sensor due to the input accelerometer freezing. (c) Uncertainty in virtual sensor due to precision degradation in the input accelerometer. (d) Change of prediction interval due to the input accelerometer freezing. (e) Gradual increase in prediction interval due to precision degradation in the input accelerometer.

prediction interval, which shows a 95% interval of ± 2 standard deviations σ around the prediction value. The prediction interval and the mean prediction are in black and red, respectively. The 95% prediction interval indicates a 5% probability the prediction will not be within the interval. We can use the *prediction interval size*, which is the magnitude of the prediction interval expressed in the unit of the target variable, to quantify the uncertainty for any given prediction.

We also examined how many actual data points from the vertical acceleration fell within the predicted range for each data point. Our results indicate that 96% of the actual data points are within the prediction interval before the onset of freezing or degradation. The 95% prediction interval can be understood as a range in which we expect to find future data points 95% of the time using our model's prediction. It is a statistical estimation that provides a range around a predicted value within which we can be 95% confident that the actual data point will fall. In

simpler terms, if you think of it as a net cast by the model, the wider this net is, the more likely it is to catch the actual data points. That is why it is significant that around 96% of actual values fell within this broad 95% prediction interval before any faults happened, like freezing or degradation. It suggests that our predictions are quite accurate.

Freezing is a phenomenon where the input sensor data suddenly becomes a constant value. We introduced freezing (at point 600 in Figure 2(b)) in our test dataset to test our virtual sensor based on a BCNN. We obtained $R^2 = -0.78$ (as opposed to $R^2 = 0.9$ without freezing) on the test dataset with freezing after training the BCNN with one random seed. This score indicates that the prediction is poor when the input is frozen. We calculated the score by comparing the virtual sensor output to the ground truth. It varies for different seeds used to train the BCNN but is still close to 0 or negative. However, it is interesting to know if one can detect uncertainty in a prediction without the ground truth. The prediction interval generated by the virtual sensor is large when freezing occurs in the input (Figure 2(b)). There is a sharp increase in the prediction interval size when there is a sensor freeze (Figure 2(d)). This unsupervised quantification of uncertainty indicates that the freezing of an input sensor is *out-of-distribution* and a source of aleatoric uncertainty. The interval experiences small changes each time we train the BCNN with a different random seed. The epistemic uncertainty in the interval is consistently high when the input sensor freezes.

Precision degradation represents a gradual change in the precision of the input accelerometer. It may occur due to changes in sensor placement emanating from the vibration the sensor is monitoring. Figure 2(c) shows the mean prediction and prediction interval during precision degradation (gradually introduced at the 600th data point). The gradual increase in the prediction interval is better observable in Figure 2(e). Like in the prediction of uncertainty during freezing, the increase in the prediction interval size helps detect aleatoric uncertainty in the sensor. The variation due to randomized training of BNNs introduces small differences in the mean predictions and interval. This epistemic uncertainty in our case study mildly affects the general trend in the aleatoric uncertainty due to precision degradation.

Implications for Trustworthy AI

The Ethics Guidelines for Trustworthy Artificial Intelligence (AI) provided by the High-Level Expert Group on Artificial Intelligence (AI HLEG) [5] list seven key requirements that AI systems should meet to be trustworthy. While answering if and how we can engineer virtual sensors that quantify the degree of their uncertainty to ensure their trustworthiness, we investigate the implication of uncertainty estimation in virtual sensors for the AI-HLEG requirements [5] concerning trustworthy AI. The discussion is meant to extrapolate from the technical contribution of generating a prediction interval and explain higher-level implications for trustworthy AI.

Human agency and oversight. Our virtual sensors are designed to guide decisions made by human end-users. They provide information to take action if a sensor fails. For instance, a large prediction interval prompts engineers to verify if the input sensor is correctly placed or not frozen. An engineer can use the mean predicted vibration in the wheel spindle head's vertical axis to stop manufacturing on the shop floor or enable vibration damping. An emergency red stop button is always available to stop a manufacturing process taking place in a cage. Preventing excessive vibration helps reduce defects in manufacturing thousands of identical parts.

Observing sensor indicators and monitoring manufacturing processes, human operators (e.g., engineers) in the manufacturing domain undergo several years of training and experience. However, understanding the output of uncertainty-aware virtual sensors is not part of their training or experience. Therefore, they need training concerning the concept of prediction interval and the root cause analysis of uncertainty.

The prediction interval in virtual sensors prevents engineers from being over-reliant on the prediction output. Any out-of-distribution input data or variability in training is reflected as a prediction interval bigger than the usual one, making engineers question the validity of the underlying ML models.

It is substantial to take specific oversights and control measures to reflect self-learning or autonomous nature of AI systems. Training BNNs involve a random component leading to epistemic uncertainty in virtual sensors. One way to address

this uncertainty is to train multiple BNNs to create an ensemble of models.

Technical robustness and safety. Technical faults, attacks, and (malicious) misuse can affect a virtual sensor as it depends on input from another physical sensor and an ML model. Inaccurate virtual sensor predictions may lead to a cascade of faults in CPS. A virtual sensor not trained on rare malicious/faulty data may give incorrect predictions for adversarial input data. Besides, an uncertainty-aware virtual sensor generates a larger prediction interval for predictions on malicious/faulty data, which warns about a potential fault, attack, or input sensor misuse.

Virtual sensors can be certified as recommended by the Cybersecurity Act [20] in Europe. Auditing the data supply chain used in creating a virtual sensor is necessary to prevent the use of malicious data. We must protect the training data from social engineering attacks coming from open datasets. Virtual sensors deployed on AI hardware (special or general purpose) are exposed to attacks. For instance, power analysis is a side-channel attack using energy consumption to derive information about what is processed on hardware. Software (virtual sensors) installed on AI hardware should be immediately patchable if required. Uncertainty estimation can help track malicious behavior and trigger a new patch.

Testing methods such as coverage-guided fuzzing, metamorphic testing, adversarial testing, and differential testing can significantly increase our confidence in virtual sensors. These methods cover testing virtual sensors beyond their training data. Prediction intervals in uncertainty-aware virtual sensors help quantify the degree of trust one may associate with adversarial inputs.

Virtual sensors may cause critical, adversarial, or damaging consequences (e.g., to human safety) in case of low reliability or reproducibility. Virtual sensors using BNNs for uncertainty estimation are affected by epistemic uncertainty due to randomization during model training. This may harm the reproducibility of the mean prediction and the prediction intervals. Aggregating the output from an ensemble of BNNs can provide a robust prediction with a high computational cost.

A low level of virtual sensor accuracy may result in critical, adversarial, or damaging consequences. We can determine the low accuracy using prediction interval. Based on the prediction interval size, we can either stop the CPS or perform continual learning on new input data. When the prediction interval increases beyond a threshold (aka the detection of concept drift), we can trigger data acquisition to train a new virtual sensor or improve the current one using techniques such as elastic weight consolidation and replay (of old data along with new data).

Transparency. This requirement encompasses three elements (i.e., traceability, explainability, and open communication) about the limitations of AI systems.

Traceability concerns having measures to continuously assess the quality of AI system output(s). The prediction interval is an inherent feature of virtual sensors to detect prediction quality. We can also use it to reveal concept drift where ML models become obsolete for changes in input data (e.g., sensor data changes due to the manufacturing process switching from producing one part to another).

Explainability refers to the ability to explain the technical processes of the AI system and the reasoning behind the decisions or predictions that the AI system makes. Uncertainty estimation is a form of explainable AI where the prediction interval tells us whether the input sensor is experiencing aleatoric uncertainty, e.g., because of cybersecurity attacks or changes in the environment. However, it cannot reveal which factor is responsible for increasing the prediction interval. Injecting faults such as freezing and precision degradation enables us to study the behavior of the prediction interval. For instance, there is a sudden increase in the prediction interval for the freezing case (Figure 2(c)), while the interval increase is gradual for precision degradation (Figure 2(d)). The patterns of uncertainty varying depending on the fault type in input data is an open area of research in explainable AI.

Users need to understand the benefits and limitations of virtual sensors. Before deploying our virtual sensors, we should demonstrate their properties, such as *simulatability*, to end-users. A model is simulatable when we can predict its behavior on new inputs. We may also study the effects of the prediction interval as an explanation in simulatability of virtual sensors, i.e., the user's ability to predict the behavior of a virtual sensor

on new data given the prediction interval.

Communication addresses the mechanisms informing users about the purpose, criteria, and limitations of the decision(s) generated by the AI system. Virtual sensor end-users always obtain a mean prediction and a prediction interval, which help them understand the limitations of the underlying ML model. We need inquiries when the prediction interval goes beyond a threshold.

Diversity, non-discrimination and fairness. A virtual sensor can be deployed into several machines operated by humans across various countries. Therefore, model training should be unbiased and fair for different manufacturing contexts that depend on the manufactured part, the shop floor, human expertise in calibrating and placing the sensors, and sensor quality. The prediction interval size is a good indicator of bias in training. Large intervals for unforeseen input sensor data indicate that the virtual sensor operates in an unforeseen context and thus requires continual training and updates for new data. This can imply that certain stakeholders in a manufacturing network better financing to maintain quality of their work.

Conclusion and Future Work

We presented an ML pipeline that generates uncertainty-aware virtual sensors based on Bayesian neural networks. We applied the pipeline to generate a virtual sensor for vibration monitoring with multiple accelerometers in automotive bearing manufacturing. We plan the following future work:

- **New Metrics**: One way forward is the inclusion of more metrics for better uncertainty estimation, e.g., feedback from user surveys, domain experts, and quantities that track overall virtual sensors performance.
- **New Test Methods:** We also need new test procedures and quality engineering methods to complement uncertainty estimation for assessing whether AI systems are trustworthy and operate as intended. Using such metrics and methods will improve the evaluation of inference scenarios and give a better picture of model reliability and the prediction process.
- **Continual Learning and Uncertainty Estimation:** Based on the uncertainty estimation, it is possible to retrain and redeploy virtual sensors continuously in a semi-automated manner [21]. We determine the uncertainty of a virtual sensor's performance in a confidence interval for the prediction. When the interval is above a threshold, we can store new data and train a new virtual sensor based on new and some old data to minimize the abrupt and drastic forgetting of previously learned information upon learning new information (*catastrophic forgetting*).

Acknowledgement

The work has been conducted as part of the InterQ project (958357) and the DAT4.ZERO project (958363) funded by the European Commission within the Horizon 2020 research and innovation programme.

REFERENCES

1. D. Martin, N. Kühl, and G. Satzger, "Virtual sensors," *Business & Information Systems Engineering*, vol. 63, no. 3, pp. 315–323, 2021.
2. Q. Sun and Z. Ge, "A survey on deep learning for data-driven soft sensors," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5853–5866, 2021.
3. S. Sen, E. J. Husom, A. Goknil, S. Tverdal, P. Nguyen, and I. Mancisidor, "Taming data quality in ai-enabled industrial internet of things," *IEEE Software*, vol. 39, no. 6, pp. 35–42, 2022.
4. J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher *et al.*, "A survey of uncertainty in deep neural networks," *arXiv preprint arXiv:2107.03342*, 2021.
5. The High-Level Expert Group on AI, "High-level expert group on artificial intelligence: Ethics guidelines for trustworthy ai," 2019. [Online]. Available: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai
6. S. Kabadayi, A. Pridgen, and C. Julien, "Virtual sensors: Abstracting data from physical sensors," in *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)*. IEEE, 2006, pp. 6–pp.
7. S. Madria, V. Kumar, and R. Dalvi, "Sensor cloud: A cloud of virtual sensors," *IEEE software*, vol. 31, no. 2, pp. 70–77, 2013.
8. M. Lee, J. Bae, and S. B. Kim, "Uncertainty-aware soft sensor using bayesian recurrent neural networks,"

Advanced Engineering Informatics, vol. 50, p. 101434, 2021.

9. W. Yan, D. Tang, and Y. Lin, "A data-driven soft sensor modeling method based on deep learning and its application," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4237–4245, 2016.
10. X. Yuan, B. Huang, Y. Wang, C. Yang, and W. Gui, "Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted sae," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 7, pp. 3235–3243, 2018.
11. X. Yuan, L. Li, and Y. Wang, "Nonlinear dynamic soft sensor modeling with supervised long short-term memory network," *IEEE transactions on industrial informatics*, vol. 16, no. 5, pp. 3168–3176, 2019.
12. X. Yin, Z. Niu, Z. He, Z. S. Li, and D.-h. Lee, "Ensemble deep learning based semi-supervised soft sensor modeling method and its application on quality prediction for coal preparation process," *Advanced Engineering Informatics*, vol. 46, p. 101136, 2020.
13. X. Wang and H. Liu, "Soft sensor based on stacked auto-encoder deep neural network for air preheater rotor deformation prediction," *Advanced engineering informatics*, vol. 36, pp. 112–119, 2018.
14. S. Sen, E. J. Husom, A. Goknil, D. Politaki, S. Tverdal, P. Nguyen, and N. Jourdan, "Virtual sensors for erroneous data repair in manufacturing a machine learning pipeline," *Computers in Industry*, vol. 149, p. 103917, 2023.
15. L. V. Jospin, H. Laga, F. Boussaid, W. Buntine, and M. Bennamoun, "Hands-on bayesian neural networks—a tutorial for deep learning users," *IEEE Computational Intelligence Magazine*, vol. 17, no. 2, pp. 29–48, 2022.
16. C. H. Lubba, S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, and N. S. Jones, "catch22: Canonical time-series characteristics," *Data Mining and Knowledge Discovery*, vol. 33, no. 6, pp. 1821–1852, 2019.
17. Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.
18. T. Papamarkou, J. Hinkle, M. T. Young, and D. Womble, "Challenges in markov chain monte carlo for bayesian neural networks," *Statistical Science*, vol. 37, no. 3, pp. 425–442, 2022.
19. D. Huseljic, B. Sick, M. Herde, and D. Kottke, "Separation of aleatoric and epistemic uncertainty in deterministic deep neural networks," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 9172–9179.
20. C. Kohler, "The eu cybersecurity act and european standards: an introduction to the role of european standardization," *International Cybersecurity Law Review*, vol. 1, no. 1, pp. 7–12, 2020.
21. S. Sen, S. M. Nielsen, E. J. Husom, A. Goknil, S. Tverdal, and L. S. Pinilla, "Replay-driven continual learning for the industrial internet of things," in *2023 IEEE/ACM 2nd International Conference on AI Engineering–Software Engineering for AI (CAIN)*. IEEE, 2023, pp. 43–55.