Bayesian autoencoders for drift detection in industrial environments

Bang Xiang Yong
Institute for Manufacturing
University of Cambridge
Cambridge, United Kingdom
bxy20@cam.ac.uk

Yasmin Fathy
Institute for Manufacturing
University of Cambridge
Cambridge, United Kingdom
yafa2@cam.ac.uk

Alexandra Brintrup
Institute for Manufacturing
University of Cambridge
Cambridge, United Kingdom
ab702@cam.ac.uk

***Abstract*—Autoencoders are unsupervised models which have been used for detecting anomalies in multi-sensor environments. A typical use includes training a predictive model with data from sensors operating under normal conditions and using the model to detect anomalies. Anomalies can come either from real changes in the environment (real drift) or from faulty sensory devices (virtual drift);however, the use of Autoencoders to distinguish between different anomalies has not yet been considered. To this end, we first propose the development of Bayesian Autoencoders to quantify epistemic and aleatoric uncertainties. We then test the Bayesian Autoencoder using a real-world industrial dataset for hydraulic condition monitoring. The system is injected with noise and drifts, and we have found the epistemic uncertainty to be less sensitive to sensor perturbations as compared to the reconstruction loss. By observing the reconstructed signals with the uncertainties, we gain interpretable insights, and these uncertainties offer a potential avenue for distinguishing real and virtual drifts.**

***Index Terms*—uncertainty, Bayesian autoencoder, deep learning, sensors**

I. Introduction

In smart factories, machine learning algorithms are increasingly used to extract values from multi-sensor data that monitor manufacturing processes whose characteristics are often complex and non-linear. In typical predictive models for manufacturing, the trust and safety in predictive models should be improved. As such, it is crucial to quantify and explain the confidence of the outcomes for the predictive models. An emerging area of research is the uncertainty quantification of deep learning models [1]. Primarily, there are two types of uncertainties, epistemic and aleatoric - the former is the uncertainty in the model parameters due to limited data availability, while the latter arises from the noise in data [2]. Nonetheless, to this end, the uncertainty of deep learning models remains understudied in most of realistic Industry 4.0 scenarios.

Due to the dynamic and ad-hoc environment of factories, the collected data by multiple sensors are often non-stationary [3]. In condition monitoring and quality prediction, machine learning (ML) methods rely on quantifying and detecting real changes in the environment and object of interest (real drift). As sensors degrade over time with increasing noise and drift level, ML methods which rely on the measurements of these sensors are affected (virtual drift).

In the occurrence of drifts (real and virtual), the noise level of the underlying distribution may change. Hence, it is important for the model to capture the change which is termed heteroscedestic aleatoric uncertainty. In contrast, in a model where the estimated noise level is assumed constant, it is called homoscedestic aleatoric uncertainty [2].

In separate studies, unsupervised deep learning models such as autoencoders were shown to perform well for detecting real drifts in applications of quality prediction [4] and prognosis [5] as well as detecting virtual drifts which are due to faults in the sensors [6]. It's crucial to distinguish between real and virtual drifts such that operators can take appropriate mitigation actions depending on the source of anomalies.

This study is the first to shed light on the behaviour of quantified epistemic and aleatoric uncertainties in autoencoders for unsupervised learning within the context of real and virtual drift in sensor data.

The paper is structured as follows. Section II provides the required background and our proposed approach. The performance evaluation including dataset, reproducibility of experimental results and evaluation are included in Section III and Section IV. We conclude the paper and explain the future directions of our research in Section V.

II. Bayesian Autoencoder

The general structure of an autoencoder aims at mapping a given set of unlabelled training data; $X = \{x_1, x_2, x_3, ...x_N\}$, $x_i \in \mathbb{R}^{\mathrm{D}}$ into an output $\hat{x}$ (i.e reconstructed signal), through a latent representation, h [7]. Structurally, every autoencoder consists of two parts: an encoder f for mapping original data x into h (i.e. $h = f(x)$) and a decoder g for mapping h to a reconstructed signal of the input $\hat{x}$ (i.e. $\hat{x} = g(h) = g(f(x))$.

Based on Bayes rule,

$$p(\theta|X) = \frac{p(X|\theta)\ p(\theta)}{p(X)} \tag{1}$$

where $p(X|\theta)$ is the data likelihood which can be modelled as a diagonal Gaussian distribution with i.i.d assumption and the likelihood mean is the Bayesian Autoencoder's output. $p(\theta)$ is the prior distribution of the Bayesian Autoencoder's parameters. For simplicity, one can assume a diagonal Gaussian prior which corresponds to an L2 regularisation. Since eq. (1)

is analytically intractable for a deep neural network which is highly non-linear and consists of a large number of parameters compared to classical statistical models, various approximate methods were developed such as Markov Chain Monte Carlo (MCMC) [8], variational inference [9], MC Dropout [10] and ensembling [11] to sample from the posterior distribution. Although these methods have been explored within supervised neural networks, to the best of our knowledge, they have not been extensively applied on autoencoders which are unsupervised models. Within these methods, the marginal distribution, $p(X)$ (or evidence) is often assumed as a constant and ignored.

In this paper, we employ a sampling method, 'anchored ensembling' [11] for approximating the posterior distribution while training the autoencoders. In anchored ensembling, posteriors are approximated by Bayesian inference under the family of methods called randomised maximum a posteriori (MAP) sampling, where model parameters are regularised by values drawn from a distribution (so-called anchor distribution), which can be set equal to the prior.

Assume our ensemble consists of M independent autoencoders and each j-th autoencoder contains a set of parameters, θ_j where $j \in \{1, 2, 3...M\}$. The autoencoders are trained by minimising the loss function, which is the negative sum of log likelihood (based on i.i.d assumption) and log prior where both are assumed to be Gaussians. The loss due to likelihood is :

$$\mathcal{L}(X, \hat{X}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{2\sigma_i^2} ||x_i - \hat{x_i}||^2 + \frac{1}{2} \log \sigma_i^2 \tag{2}$$

where σ_i^2 is the variance of the data point, which is also known as aleatoric uncertainty for regression tasks. Note that a typical autoencoder minimises the reconstruction loss, $||x_i - \hat{x_i}||^2$ which corresponds to a diagonal Gaussian likelihood with fixed variance of 1. Instead of a fixed variance, by 'learning' the variance term as an output of the autoencoder, the model can estimate the noise level for every data point x_i. Following the method proposed by [2] to compute heteroscedastic aleatoric uncertainty for regression tasks, an extra layer is added to the final layer of autoencoder with dimensions equal to the size of the inputs, to predict the log variance, $\log \sigma_i^2$ corresponding to each data point x_i.

The 'anchored weights' for each autoencoder are unique and sampled during initialisation from a prior distribution $\theta_{anc,j} \sim N(\mu_{anc,j}, \sigma_{anc,j}^2)$ and remain fixed throughout the training procedure. To scale the regulariser term arising from the prior, λ is set as a hyperparameter. The loss due to prior is:

$$\mathcal{L}(\theta_j) = \frac{\lambda}{N} \sum_{i=1}^{N} ||\theta_j - \theta_{anc,j}||^2 \tag{3}$$

With eq. (2) and eq. (3), the resulting loss function to be minimised is:

$$\mathcal{L}(X, \hat{X}, \theta_j) = \mathcal{L}(X, \hat{X}) + \mathcal{L}(\theta_j) \tag{4}$$

For model prediction, the predictive posterior distribution of an unseen test input X^*, is calculated by integrating over all possible θ:

$$p(X^*|X) = \int p(X^*|\theta, X)\ p(\theta|X)\ d\theta \tag{5}$$

Although eq. (5) is intractable, we can estimate it with the samples of $p(\theta|X)$ which we obtained by training the ensemble:

$$\hat{p}(X^*|X) = \sum_{j=1}^{M} p(X^*|\theta_j, X) \tag{6}$$

To compute epistemic uncertainty on a new single data point, x^*, the variance of reconstructed signals from the ensemble is computed:

$$Var(\hat{x}^*) = \frac{\sum_{j=1}^{M} (\hat{x}_j^* - \bar{x})^2}{M} \tag{7}$$

where M is the number of ensembled autoencoders, $\bar{x}$ is the mean of reconstructed signals $\hat{x}_j^*$.

In addition to the reconstructed signals, the Bayesian Autoencoder also outputs the log variance of data, $\log \sigma_i$, by which we can recover the heteroscedestic aleatoric uncertainty, σ_i^2 with the exponential function.

III. Experimental Evaluation

This section explains the real-world dataset used in our evaluation, the reproducibility of our results and the evaluation criteria of our proposed method. We discuss the results in the following section.

A. Dataset

We have tested our proposed approach on a publicly available dataset for condition monitoring of hydraulic system [12]. The dataset is obtained from a hydraulic test rig which permits safe and non-destructive changes to the states of various components (cooler, valve, pump and accumulator) to emulate faults and degradation. Redundant sensors are equipped on the test system on multiple locations to measure pressure, flow, temperature, power and vibration. There is a total of 17 sensors and various sampling frequencies of 1Hz, 10Hz and 100Hz. In the dataset, there is a total of 2205 cycles and each working cycle of the hydraulic system lasts for 60 seconds. The methods developed in our study are not limited to condition monitoring and can be applied to other Industry 4.0 use cases.

B. Data processing

Due to the inconsistent sampling frequencies, we resample the data to 1Hz. As such, this results in 60 (time points) * 17 (sensors) for each cycle. The features are then normalised using a standard scaler for each sensor with careful implementations to prevent train-test bias. We do not compute specific features from the data but instead we feed the resampled and rescaled raw signals to the Bayesian Autoencoder. By doing so, we are able to visualise and gain insights into the full reconstructed signals as predicted by the deep model.

C. Experiment setup

We set the number of hidden nodes and layers of the Bayesian Autoencoder to 1020-500-250-3-250-500-1020 with 10 samples in the ensemble. The Bayesian Autoencoder is trained and tested with 70% and 30% respectively of the sensor data where the cooler condition is known to be healthy. Then, for the case of real drift, we test the model on data which the cooler condition has degraded to 20% and 3% (near failure) efficiency. To simulate virtual drifts scenarios, we create two datasets from the 'healthy' test set and artificially inject a range of noise from 5-25% (i.e injected noise of uniform distribution) and constant sensor drift of 5-25% of the mean in each one of the sensors (i.e injected drift).

To ensure the reproducibility of our results, we have made the code of our implementation available and have also provided details of a configurable experimental set-up at https://github.com/bangxiangyong/bae-drift-detection-zema-hydraulic.

IV. RESULTS AND DISCUSSION

We have conducted three sets of experiments; 1) real drift, 2) injected noise, and 3) injected drift. The reconstruction loss, epistemic and aleatoric uncertainties for these experiments are summarised in Fig. 1, Fig. 2 and Fig. 3 respectively. Although we show the results for only one of the pressure sensors, denoted PS1, we have extended the experiment to every sensor and found similar results. One limitation of our analysis is we do not explore virtual drifts on combination of sensors. In general, we note that the mean of reconstruction loss, epistemic uncertainty and aleatoric uncertainty increase in both cases of real and virtual drift conditions.

The reconstruction loss for the cooler condition of %3 has a longer tail which overlaps with the less faulty conditions (Fig. 1a). In practice, when using it for anomaly detection, this may lead to false positives. Additionally, the reconstruction loss and aleatoric uncertainty increase exponentially with the degrading condition of cooler, whereas epistemic uncertainty increases linearly in the same scenarios (Fig. 1).

Moreover, the epistemic uncertainty is generally less affected by noise in the sensor than the reconstruction loss (Fig. 2). Unexpectedly, however, in the advent of increasing sensor noise, the aleatoric uncertainty does not increase as shown in Fig. 2c. Intuitively, we expect the estimated variance to be proportional to the level of sensor noise. In contrast, we note that the aleatoric uncertainty increases dramatically for the degrading cooler condition, since multiple sensors are affected simultaneously. Therefore, comparing these two situations, an exploration step is to investigate the effects of perturbations applied on a combination of sensors. With that, we can develop a feature importance ranking based on the sensitivity of the model's uncertainties.

In the case of injected sensor drift (Fig. 3), the reconstruction loss increases exponentially, whereas the epistemic uncertainty increases almost linearly. In contrast, aleatoric uncertainty shows a convex behaviour. Unfortunately, since the aleatoric uncertainty is computed using a black-box model, we do not have an intuitive explanation for it [13].

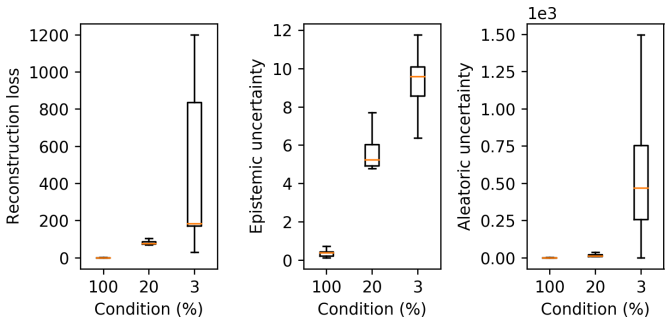

Fig. 1. a) Reconstruction loss, b) Epistemic uncertainty, c) Aleatoric uncertainty under real drift of degrading cooler condition

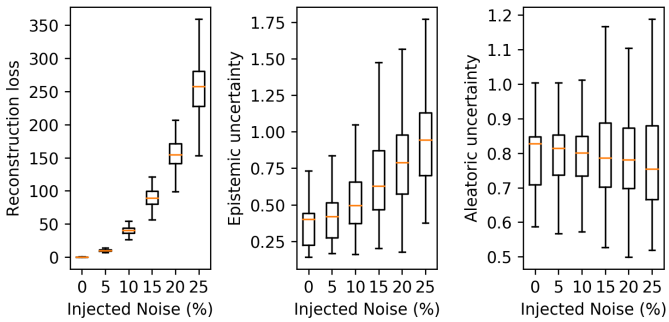

Fig. 2. a) Reconstruction loss, b) Epistemic uncertainty, c) Aleatoric uncertainty under virtual drift of increasing noise in a pressure sensor

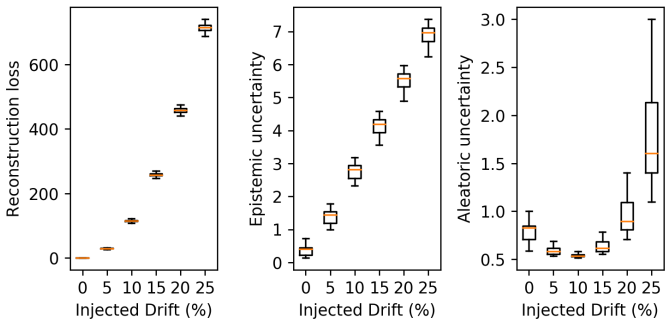

Fig. 3. a) Reconstruction loss, b) Epistemic uncertainty, c) Aleatoric uncertainty under virtual drift of increasing drift in a pressure sensor

By solely relying on the reconstruction loss, we are unable to distinguish real and virtual drifts. Thus, we posit that, by capturing these patterns of uncertainties, novel methods can potentially be developed to distinguish real and virtual drifts in sensor data as shown in Fig. 4. From a qualitative perspective, we note that the points form clusters which are separable. This implies we can apply a clustering algorithm (e.g k-means or hierarchical clustering) on these three metrics: reconstruction

loss, epistemic uncertainty and aleatoric uncertainty of every data point to achieve unsupervised classification. To the best of our knowledge, we are the first to elicit this application within Bayesian Autoencoders.

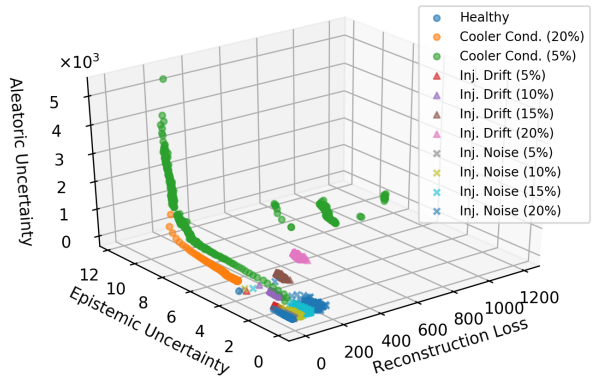

Fig. 4. Scatter plot of Bayesian Autoencoder's outputs under various conditions: healthy condition, degrading cooler condition, noisy and drifting pressure sensor. This illustrates the separability of the types of drifts based on the trio: reconstruction loss, epistemic uncertainty and aleatoric uncertainty.

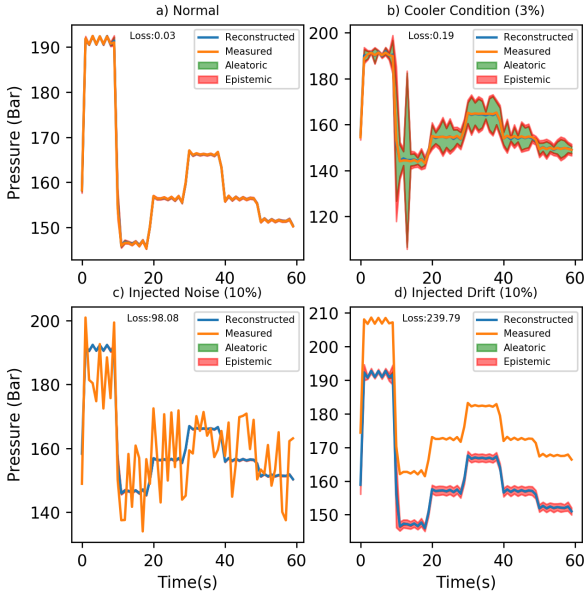

Fig. 5. Measured signal and reconstructed signal of pressure sensor with epistemic and aleatoric uncertainties in a working cycle

We have conducted further experiments (in Fig. 5) to gain more insights about the actual, reconstructed values and their uncertainties. For the nearly faulty cooler condition (Fig. 5b), the reconstruction loss shows an insignificant increase compared to the normal actual signal (Fig. 5a). However, the epistemic and aleatoric uncertainties increase significantly. Despite the presence of noise and drifts (Fig. 5c & d), we note that the Bayesian Autoencoder is able to reconstruct the shape of the normal signal. In such a case, the reconstruction loss increases rapidly; this is due to the high difference between the actual and reconstructed values. Meanwhile, the uncertainties do not show significant increase in these situations. By observing the uncertainties of the reconstructed signal, operators can gain more interpretable insights into the model's predictions. Since the uncertainties are computed on a feature level, the uncertainty of every sensor on every time step can be leveraged for further decision making.

Future work will involve using a Gaussian likelihood with a full covariance matrix, instead of a diagonal only (as in this experiment), which may reveal more insights in interpreting the model's aleatoric uncertainty measures. Other than a Gaussian likelihood, the effects of using different likelihood distributions can also be explored. Moreover, we can leverage the Bayesian Autoencoder's outputs for a novel unsupervised classification method. We will also extend the experiments to study the effect of variant Bayesian Autoencoder architectures and various datasets on identifying the real and virtual drifts and their uncertainties.

V. Conclusion

Distinguishing between a real and virtual drift is of importance, especially in ML for manufacturing where the environments are highly dynamic. Our conducted experiments show that the reconstruction loss typically used in autoencoders is unable to distinguish a real drift in the environment and virtual drift due to sensor degradation. By observing the epistemic and aleatoric uncertainties, a difference is noticed in the quality of prediction in each case, which can be leveraged for distinguishing real and virtual drifts in sensors data. Since uncertainty quantification using Bayesian Autoencoders is largely unexplored in the industrial context, this appears to be a promising field of research which deepens our understanding and trust of these deep models. We leave the detailed analysis of these observations for future studies.

Acknowledgment

The research presented was supported by European Metrology Programme for Innovation and Research (EMPIR) under the project Metrology for the Factory of the Future (MET4FOF), project number 17IND12 as well as the PITCH-IN (Promoting the Internet of Things via Collaborations between HEIs and Industry) project funded by Research England. We express our gratitude to Tim Pearce for his inputs.

References

[1] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.

[2] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[3] B. X. Yong and A. Brintrup, "Multi agent system for machine learning under uncertainty in cyber physical manufacturing system," in *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing*. Springer, 2019, pp. 244–257.

[4] G. Wang, A. Ledwoch, R. M. Hasani, R. Grosu, and A. Brintrup, "A generative neural network model for the quality prediction of work in progress products," *Applied Soft Computing*, vol. 85, p. 105683, 2019.

[5] M. Martinez-Garcia, Y. Zhang, J. Wan, and J. Mcginty, "Visually interpretable profile extraction with an autoencoder for health monitoring of industrial systems," in *2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM)*. IEEE, 2019, pp. 649–654.

[6] J.-W. Yang, Y.-D. Lee, and I.-S. Koo, "Convolutional autoencoder-based sensor fault classification," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2018, pp. 865–867.

[7] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning book," *MIT Press*, vol. 521, no. 7553, p. 800, 2016.

[8] T. Chen, E. Fox, and C. Guestrin, "Stochastic gradient hamiltonian monte carlo," in *International Conference on machine learning*, 2014, pp. 1683–1691.

[9] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *International Conference on Machine Learning*, 2015, pp. 1613–1622.

[10] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *International Conference on Machine Learning*, 2016, pp. 1050–1059.

[11] T. Pearce, M. Zaki, A. Brintrup, N. Anastassacos, and A. Neely, "Uncertainty in neural networks: Bayesian ensembling," *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

[12] N. Helwig, E. Pignanelli, and A. Schütze, "Condition monitoring of a complex hydraulic system using multivariate statistics," in *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, May 2015, pp. 210–215.

[13] B. Venkatesh and J. J. Thiagarajan, "Heteroscedastic calibration of uncertainty estimators in deep learning," *arXiv preprint arXiv:1910.14179*, 2019.