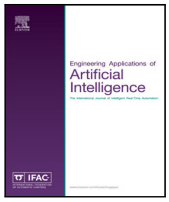
Engineering Applications of Artificial Intelligence 127 (2024) 107233

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Predicting product quality in continuous manufacturing processes using a scalable robust Gaussian Process approach ☆

Diego Echeverria-Rios, Peter L. Green *

School of Engineering, The University of Liverpool, L69 3GH, UK

ARTICLE INFO

Keywords:
Manufacturing
AI
Foundation industries
Gaussian process
Robust

ABSTRACT

This work describes an Artificial Intelligence (AI)-based solution that predicts product quality when applied to a continuous manufacturing process. The proposed solution uses process parameters and product quality measurements that are obtained from a production line. The work detailed herein is problem-driven, showing an application within one of the UK's foundation industries and identifying five key criteria an AI solution should ideally satisfy in continuous manufacturing applications; scalability, modularity, stable out-of-data performance, uncertainty quantification and robustness to unrepresentative data. The shortcomings, relative to these five criteria, of available AI approaches are discussed before a potential solution is presented. The proposed approach involves the application of a generalised product-of-expert Gaussian process whose noise model is constructed from a Dirichlet process. The ability of the model to fulfil the five key criteria and its performance when applied to the foundation industry case study is demonstrated.

1. Introduction

In this paper we consider a foundation-industry application whereby a product is developed from a continuous manufacturing process. Data regarding process parameters is collected at various points during the product line while product quality (specifically, the number of faults per unit area a.k.a. 'fault density') is measured at the end of the production line. The aim is to develop an 'AI' (a.k.a 'machine-learnt' or 'data-based') model that can be used to predict product quality as a function of process parameters and, subsequently, be used to optimise the manufacturing process. From previous research (Liu et al., 2018b; Feng et al., 2009; Jin et al., 2020) and based on the authors' experience regarding the development of AI solutions and their deployment within the continuous manufacturing application space, several key criteria must be considered from the very start of the model development process:

1. Scalability. A continuous manufacturing process will, by definition, generate a continuously growing set of data. The proposed AI solution must therefore be scalable to large datasets.
2. Modularity. Data has a lifecycle; one can expect that, as a result of changes in operation, product etc., old data will become less representative of current operation as time passes. It must therefore be possible to remove/augment the information in this data in a modular fashion i.e. without stopping the overall AI model from functioning.
3. Stable out-of-data performance. When applied far from the region covered by the training data the model must 'fail gracefully', illustrating to the user that the production line is currently in a state where the model should not be used to influence decision making. Such scenarios can occur, for example, after a sensor failure leads to erroneous measurements of manufacturing process parameters or if a new product, not previously included in the training data, is being produced.
4. Uncertainty quantification. The model should reflect the confidence that it has in its predictions; another key component regarding its suitability as a decision-making aid.
5. Robustness to unrepresentative data. The data used to train an AI model must, by definition, be representative of the process of interest; unrepresentative data (data over periods where the manufacturing process is affected by factors outside the scope of the model e.g. site repairs, unmeasured changes in raw materials) must be excluded during training. The identification of these unrepresentative data can, however, be difficult; particularly when the affects of un-modelled external factors are hidden amongst the general variability of the manufacturing process.

☆ **Funding:** The authors acknowledge support from the Engineering and Physical Sciences Research Council grant: Transforming the Foundation Industries Network+ EP/V026402/1.

* Corresponding author.
E-mail address: p.l.green@liverpool.ac.uk (P.L. Green).

https://doi.org/10.1016/j.engappai.2023.107233
Received 14 June 2023; Received in revised form 19 September 2023; Accepted 28 September 2023
Available online 12 October 2023

Having established these 5 key criteria it is possible to perform a pre-implementation evaluation of existing approaches, based on their suitability. Artificial Neural Networks (ANNs), for example, are known to exhibit erratic behaviour when extrapolated beyond the training data (see e.g. Alcorn et al. (2019) and discussion in Marcus (2020)) whilst the computational cost of performing uncertainty quantification can also be prohibitive, particularly for so-called 'Deep' ANNs that may have many millions of uncertain parameters. Approaches introduced in the context of Robust Parameter Design (Zhou et al., 2021; Feng et al., 2021; Zhou et al., 2023) (a methodology used to improve the quality of products and processes) can be analysed in a similar manner. Ref. Zhou et al. (2021) uses a Sequential Support Vector Regression approach that can incorporate additional data points without needing to retrain over all previous data (satisfying criterion 2) but does not scale well to large data sets, while its performance in the presence of outlier data is not discussed. Ref. Feng et al. (2021) uses a Gaussian Process (GP) with a student-t likelihood to address outliers, provide stable out-of-data performance and facilitate uncertainty quantification, but a strategy for managing incoming data is not described and the approach does not scale well to large datasets. Specifically, examples described within Feng et al. (2021) use only 10s of training points and, moreover, utilise Markov Chain Monte Carlo whose computational cost can become prohibitive. Ref. Zhou et al. (2021) describes a Gaussian Process approach that can process new data in an online fashion, but does not scale to large datasets (the largest example problem detailed in Zhou et al. (2021) extends to only 1000 training points) or have a strategy for addressing outlier data.

Aside from the key 5 criteria described above, the authors' note that non-parametric solutions have proved to be favourable, if not strictly necessary, for the current application. Whilst approaches such as Partial Least Squares and variants thereof (e.g. Principal Component Regression, Regularised Least Squares) are used in many industrial case studies (e.g. Qin et al. (2022)), they are linear regression approaches that require the specification of a parametric family of regressors (i.e. basis functions) before the training set is observed. Obtaining a suitable choice of basis function can be problematic and is known to be a deficiency of such approaches (Bishop, 2006). As a result, and following other work on the application of AI in industrial processes (Zhou et al., 2021; Feng et al., 2021; Zhou et al., 2023), we have elected to only consider non-parametric regression approaches in the current paper.

In the present work, we propose a robust Gaussian Process approach that uses a Dirichlet Process (DP) mixture-of-Gaussian distributions for the identification of different noise processes that corrupt fault density data. The proposed GP employs local computations in the regression step to overcome the standard GP memory issues that arise when using large training datasets (Tresp, 2001; Titsias, 2009; Deisenroth and Wei Ng, 2015; Liu et al., 2018a) and, as such, is scalable to larger datasets (criterion 1). We modularise the proposed method using an assembly of model experts, where each expert is a robust GP; each expert can easily be removed/re-trained (criterion 2) while, by inheriting properties associated with standard Gaussian Processes, also satisfy the requirements for stable out-of-data performance (criterion 3) and uncertainty quantification (criterion 4). Finally we note that, through the use of the mixture-of-Gaussian likelihood, the proposed methodology can be used to automatically identify and exclude data that is not representative of the process being modelled (criterion 5).

The paper is organised as follows. Gaussian Process are reviewed in Section 2, while Section 3 describes the adoption of a non-Gaussian noise model within a Gaussian Process framework. Section 4 describes the general model implementation. Section 5 describes a numerical case study involving synthetic data, while Section 6 details the application of our approach to data from one of the UK's foundation industries. Finally, Section 7 describes future work before conclusions are drawn in Section 8.

2. Gaussian processes

2.1. Relevant literature

Gaussian Processes (GPs) are a widely used machine learning technique that can be applied to both classification and regression problems. In this paper, we focus on the use of GPs as a regression tool.

GP regression is a probabilistic approach that aims to infer a latent function from observed data. An advantageous characteristic of GPs lies in their ability to quantify the uncertainty associated with their predictions. Furthermore, rather than inferring parameters of functions, a GP samples directly from a distribution over functions — for this reason, GPs are considered to be non-parametric models since they are not restricted to a specific parametric family of regressors. Successful applications of GPs can be found in multiple disciplines, such as, traffic flow (Sun and Xu, 2011), engine modelling (Chati and Balakrishnan, 2017), structural dynamics (Worden and Green, 2016), robotics (Deisenroth et al., 2015) and more.

Standard GPs rely on the assumption that the training data has been corrupted with noise drawn from a Gaussian distribution. Works such as Stegle et al. (2008), Lázaro-Gredilla and Titsias (2011) and Zhu et al. (2018), however, highlight real-world examples where the Gaussian observation model does not accurately represent reality. Neal (1997) illustrates how the accuracy of standard GP predictions can be affected when the training data has been corrupted with noise drawn from a non-Gaussian distribution. GP approaches with a likelihood derived from a student-t observation model have been proposed to help ignore the contribution of outliers (e.g. Feng et al. (2021)). In such cases, the marginal likelihood is analytically intractable, and hence, approximate methods such as, Markov chain Monte Carlo (MCMC) (Neal, 1997), variational techniques (Kuss, 2006), Laplace approximations (Vanhatalo et al., 2009) and the Expectation-Propagation (EP) method (Jylänki et al., 2011) have been applied to facilitate parameter estimation. GP models that use a heavy-tailed observation model to eliminate the contribution of outliers in the training data are sometimes referred to as 'robust GPs' (Vanhatalo et al., 2009) (terminology that is also adopted for the current paper).

In 2001, Tresp (2001) proposed the so-called Mixture of Gaussian Processes (MGP) model, which is a variant of the Mixture of Experts model (Jacobs et al., 1991). The MGP model assumes that each observation has been corrupted independently by Gaussian noise, whose variance is constant only across separate regions of the input space. Accordingly, a single GP is assigned to each of these regions, and a gating function activates the corresponding GP according to the noise model that applies in that region. Aiming to infer latent heart rate time series, Stegle et al. (2008) proposed a different approach to that of the MGP model. Given that heart rate data collected during non-laboratory conditions is known to contain outliers and 'noise bursts' (Stegle et al., 2008), Stegle proposed a 2-step model based on the iterative application of unsupervised clustering and GP regression. The methodology was developed based on the assumption of a mixture of Gaussian distribution observation model. In the clustering component of the approach, the data association problem[1] (Murphy, 2012; Bar-Shalom et al., 1990; Cox, 1993) was addressed to estimate the noise structure. The number of clusters associated with the noise levels (i.e. the number of components in the mixture of Gaussian observation model) was determined by evaluating the model evidence. The cluster associated with the lowest noise level was later used to train a GP. Finally, the Expectation-Propagation method was used to approximate the predictive distribution. The main difference between Stegle's model and the MGP lies in the fact that Stegle's approach does not use a gating function, rather, a single GP is used.

[1] The data association problem focuses on inferring groups of data that originated from the same source.

In Lázaro-Gredilla et al. (2011) the Overlapping Mixture of Gaussian Processes (OMGP) model was proposed to address the data association problem in multi-object target tracking problems.[2] The authors aimed to cluster observations into trajectories, such that each trajectory could then be associated with a separate GP. A variational Bayesian inference approach was used for parameter estimation, based on the assumption that the number of trajectories is known. Motivated by the need to identify new patterns associated with lung diseases, Ross and Dy (2013) extended Lazaro-Gredilla's approach to a fully non-parametric Bayesian model where, by using a Dirichlet Process (DP) mixture of GPs, the number of trajectories associated with lung diseases was determined directly from the training data.

The previously mentioned contributions (Stegle et al., 2008; Neal, 1997; Kuss, 2006; Vanhatalo et al., 2009; Jylänki et al., 2011; Lázaro-Gredilla et al., 2011; Ross and Dy, 2013) address scenarios where training data is corrupted with non-Gaussian noise, however, another problem associated with standard GP regression is the poor scalability with respect to the number of training points, N. Specifically, the cubic complexity training time, $\mathcal{O}(N^3)$, of a standard GP often makes the approach intractable for datasets with size $N > 10^4$ (Deisenroth and Wei Ng, 2015; Liu et al., 2019) (though recent developments have pushed this envelope further Wang et al., 2019). To address this issue, scalable GP models have been developed for applications involving large datasets. Scalable GPs can be grouped into two categories; those that realise global approximations and those that realise local approximations. Global approximation approaches use relatively small sets of 'inducing points' to summarise the information that is contained in the full set of training data (variational sparse GPs Titsias, 2009, for example, belong to this category). On the other hand, local approximations use GP experts that are trained on local subsets of the training data (Tresp, 2001; Deisenroth and Wei Ng, 2015; Liu et al., 2018a). In the present contribution, motivated by a manufacturing case study where the assumption of a Gaussian noise model was found to be poorly suited to measurements of product quality, we address GP regression problems where the measurement noise is non-Gaussian. We assume that the noise corrupting the observations has been generated from a mixture of Gaussian distributions and propose a 2-step method based on DP clustering and GP regression. Unlike Stegle's approach (Stegle et al., 2008), where the number of clusters is determined by evaluating the model evidence with respect to different numbers of clusters, our approach uses a DP mixture of Gaussian distributions for clustering, which is non-parametric in the number of mixture components; this differs from Feng et al. (2021), for example, where the parametric form of the likelihood (student-t) is chosen *a-priori*. We use a variational Bayesian inference approach (Blei and Jordan, 2006a) to determine the hidden variables involved in the clustering step. Using the observations associated with the lowest noise component (i.e. those identified as being corrupted by noise that has been generated from the Gaussian whose variance is the smallest amongst the mixture), we recover the standard GP log-likelihood for model training. Subsequently, the inferred model parameters, i.e. the GP hyperparameters and the noise model parameters, are used together in the OMGP predictive distribution (Lázaro-Gredilla et al., 2011). Finally, by distributing the GP computations (Deisenroth and Wei Ng, 2015), we illustrate that the proposed approach is scalable in the number of training points, a characteristic that is usually compromised in robust GP models e.g. Stegle et al. (2008), Jylänki et al. (2011) and Liu et al. (2019).

[2] In multi-object tracking problems, a trajectory refers to the path or paths described by the observations associated with the object(s) being tracked (Murphy, 2012).

2.2. Standard Gaussian process

In the following, training data consists of a collection of input–output pairs $\{\mathbf{x}_n \in \mathbb{R}^D, y_n \in \mathbb{R}\}_{n=1}^N$. Each y_n is considered to be a noisy observation of a latent function, $f(\mathbf{x}_n)$ (in the context of the current paper, for example, y_n is the nth observation of fault density). With a standard GP approach, it is assumed that the noise corrupting each observation is sampled from a zero-mean Gaussian distribution with variance σ^2 such that

$$y_n = f(\mathbf{x}_n) + \epsilon_n \qquad \epsilon_n \sim \mathcal{N}\left(0, \sigma^2\right) \tag{1}$$

The set of observations $\boldsymbol{y} = \{y_n\}_{n=1}^N$ are a realisation of the stochastic process defined by Eq. (1) at inputs $\boldsymbol{X} = \{\mathbf{x}_n\}_{n=1}^N$. With a GP we define a prior over the function values $\boldsymbol{f} = \{f_n\}_{n=1}^N$:

$$p(\boldsymbol{f}) = \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \boldsymbol{K}) \tag{2}$$

where each element of $\boldsymbol{K}$ is defined by a *kernel function*, which ensures that $\boldsymbol{K}$ is symmetric and positive-semidefinite (Rasmussen, 2006). One such kernel function is the Squared Exponential (SE) kernel,

$$k_{\text{SE}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left\{-\frac{(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}{2\ell^2}\right\} \tag{3}$$

where parameters σ_f and ℓ are usually referred to as the *vertical length scale* and the *horizontal length scale*, respectively. Another well-known covariance function that recovers the SE kernel as a special case is the Matérn kernel,

$$k_{\text{Mat}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}{\ell}\right)^{\nu} \times K_{\nu}\left(\frac{\sqrt{2\nu}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)}{\ell}\right) \tag{4}$$

where, Γ is the Gamma function, K_{ν} is the modified Bassel function of the second kind, σ_f, ℓ are the process standard deviation and length scale, respectively, and ν controls the smoothness of the sample functions. When assigning $\nu \to \infty$ the SE kernel is recovered.

Kernels can be extended to incorporate a length scale for each input. For instance, the SE kernel can be expressed as follows,

$$k_{\text{SE}}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T \boldsymbol{L}(\mathbf{x}_i - \mathbf{x}_j)\right\} \tag{5}$$

where $\boldsymbol{L} = \text{diag}(l_1^{-2}, \ldots, l_D^{-2})$.

Eq. (5) represents an implementation of Automatic Relevance Determination (ARD), as the inverse of the length scale indicates the relevance of the corresponding input. Specifically, as the ith length scale increases, the more insensitive the GP will become to changes in the ith input. Eqs. (3) and (5) are just a few examples of the widely variety of kernels or combination of kernel functions that can be used for different applications (Duvenaud, 2014).

Having defined a kernel function then, from Bayes' theorem, we obtain

$$p(\boldsymbol{f}|\boldsymbol{y}) \propto p(\boldsymbol{f})p(\boldsymbol{y}|\boldsymbol{f}) \tag{6}$$

where $p(\boldsymbol{f})$ is the GP prior specified in Eq. (2) and, from Eq. (1), $p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{I}\sigma^2\right)$. The marginalised likelihood can be obtained by integrating over $\boldsymbol{f}$, from which we find that $p(\boldsymbol{y}) = \int \mathcal{N}(\boldsymbol{f}|\mathbf{0}, \boldsymbol{K})\, \mathcal{N}\left(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{I}\sigma^2\right) d\boldsymbol{f} = \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \boldsymbol{C})$, where $\boldsymbol{C} = \boldsymbol{K} + \boldsymbol{I}\sigma^2$, such that $C(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) + \delta_{ij}\sigma^2$ where δ_{ij} is equal to 1 if $i = j$ and 0 otherwise. Including σ as a parameter to be estimated, we define $\boldsymbol{\theta} = \{\sigma_f, l_1, \ldots, l_D, \sigma\}$ and write the likelihood of $\boldsymbol{\theta}$ as follows

$$p(\boldsymbol{y}|\,\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{y}|\mathbf{0}, \boldsymbol{C}) \tag{7}$$

Taking the logarithm of Eq. (7) one obtains

$$\ln p(\boldsymbol{y}|\,\boldsymbol{\theta}) = -\frac{1}{2}\ln|\boldsymbol{C}| - \frac{1}{2}\boldsymbol{y}^T\boldsymbol{C}^{-1}\boldsymbol{y} - \frac{N}{2}\ln(2\pi) \tag{8}$$

The process of finding the parameters, θ, that maximises the log-likelihood function is called Maximum Likelihood Estimation (MLE). This can be achieved, for example, using Gradient Based Methods (Arora, 2006) applied to Eq. (8).

3. Proposed noise model

From Section 2.2 it can be seen that the standard GP implementation is based upon the assumption of a Gaussian noise model (i.e. Eq. (1)). While this assumption can be justified to a certain extent for the generic case (using, for example, the Central Limit Theorem or the Principle of Maximum Entropy), the assumption of a Gaussian noise model was found to be detrimentally inaccurate for the case study of interest, where the goal is to predict the fault density of the final product from a foundation industry application.

Aiming to still take advantage of the closed-form solutions associated with standard GP models we adopt a Gaussian mixture observation model that is centred on the latent function. Specifically, we assume that the noise corrupting each observation has been generated from one of K Gaussian distributions. By introducing *1-of-K* allocation variables, $\boldsymbol{z}_n \in \mathbb{R}^K, n = 1,\dots,N$, with $\{z_{nk} \in \{0,1\} \,|\, \sum_{k=1}^{K} z_{nk} = 1, \forall n\}$, we associate each observation with a single Gaussian from the mixture (e.g. $z_{nk} = 1$ indicates that the observation y_n was corrupted by noise drawn from the kth Gaussian[3]). Following a Bayesian framework, we place priors on the allocation variables, $\boldsymbol{z}_n$, as follows

$$p(z_{nk} = 1) = \pi_k \qquad \text{where} \quad \sum_{k=1}^{K} \pi_k = 1 \tag{9}$$

where $\pi_1,\dots,\pi_K$ are known as the *mixture proportionalities*. Notice that we have followed the notation described in Bishop (2006), where p is used to describe both discrete and continuous probability distributions. Marginalising the joint distribution $p(\boldsymbol{z}_n)p(\epsilon_n|\boldsymbol{z}_n)$ over the possible states of $\boldsymbol{z}_n$:

$$p(\epsilon_n) = \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n)p(\epsilon_n|\boldsymbol{z}_n) \tag{10}$$

the non-Gaussian observation model can be obtained by substituting Eq. (9) into Eq. (10) and by defining $p(\epsilon_n|z_{nk} = 1)$ as a Gaussian with zero mean and variance σ_k^2, such that

$$y_n = f(\mathbf{x}_n) + \epsilon_n, \qquad \epsilon_n \sim \sum_{k=1}^{K} \pi_k \, \mathcal{N}\left(0, \sigma_k^2\right) \tag{11}$$

A wide variety of heavy tailed distributions, centred on the latent function, can be described by varying the number of components K and the parameters of Eq. (11).

We now aim to derive an expression that describes the probability of witnessing the observed data as a function of the mixture parameters described in the observation model, Eq. (11). Noting that

$$p(y_n|f_n, z_{nk} = 1) = \mathcal{N}\left(y_n|f_n, \sigma_k^2\right) \tag{12}$$

and that $p(y_n|f_n, z_{nk} = 1) = p(\epsilon_n|z_{nk} = 1) = \mathcal{N}\left(\epsilon_n|0, \sigma_k^2\right)$ allows us to write

$$p(y_n|f_n, z_{nk} = 1) = \mathcal{N}\left(r_n|0, \sigma_k^2\right) \tag{13}$$

where $r_n = y_n - f_n$ is the nth residual. Assuming that the noise corrupting each observation is independent and identically distributed (*iid*), the likelihood of witnessing the measurements of product quality data is given by

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{Z}, \boldsymbol{\sigma}) = \prod_{n=1}^{N}\prod_{k=1}^{K} \mathcal{N}\left(r_n|0, \sigma_k^2\right)^{z_{nk}} \tag{14}$$

where $\boldsymbol{Z} = \{\boldsymbol{z}_n\}_{n=1}^{N}$ and $\boldsymbol{\sigma} = \{\sigma_k\}_{k=1}^{K}$. An iterative approach for the estimation of r_n, σ_k^2 and z_{nk} is described in the next section.

[3] For more information the reader is referred to introductory material on Gaussian mixture models (e.g. Bishop (2006)).

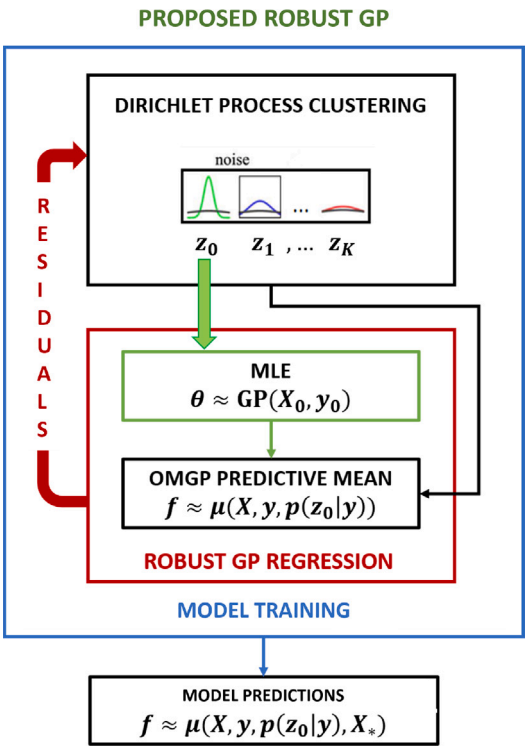


Fig. 1. Proposed robust GP diagram.

4. Model implementation

We propose to estimate the residuals, $\boldsymbol{r} = \boldsymbol{y} - \boldsymbol{f}$, in Eq. (14) by approximating $\boldsymbol{f}$ through GP regression. We, therefore, estimate the parameters in Eq. (14) by iteratively applying clustering and regression as follows:

- **Clustering:** We infer the mixture parameters $(\boldsymbol{Z}, \boldsymbol{\sigma})$ of Eq. (14) using a variational approximation (Blei and Jordan, 2006b) that is based on a Dirichlet process.
- **Regression:** We perform GP regression using only the observations identified as being corrupted with the lowest noise component. Once an estimate of the GP hyperparameters has been obtained, we realise new approximations of $\boldsymbol{f}$ using a predictive distribution that incorporates the full training dataset $\{\mathbf{x}_n \in \mathbb{R}^D, y_n \in \mathbb{R}\}_{n=1}^{N}$ and the allocation variable posterior distribution $p(\boldsymbol{z}_k|\boldsymbol{y})$.

For large scale applications, a Product-of-Experts (PoE) GP is used to realise relatively cheap approximations of $\boldsymbol{f}$. In the following, for example, a PoE-GP is used whereby the aggregation of the experts' predictions is performed using the generalised Product of Experts (gPoE) approach (Deisenroth and Wei Ng, 2015).

Fig. 1 and Fig. 2 show, respectively, a summary diagram of the proposed robust and scalable robust GPs, where the 2-main steps, clustering and regression, and their corresponding dependencies are illustrated.

4.1. The clustering step: Dirichlet process mixtures

In the current section, clustering of the residuals is performed using a DP mixture of Gaussian distributions. A DP is a stochastic process whose indexed random variables are a collection of probability measures that sum up to one with probability one. This means that a realisation from a DP is a random probability distribution, denoted

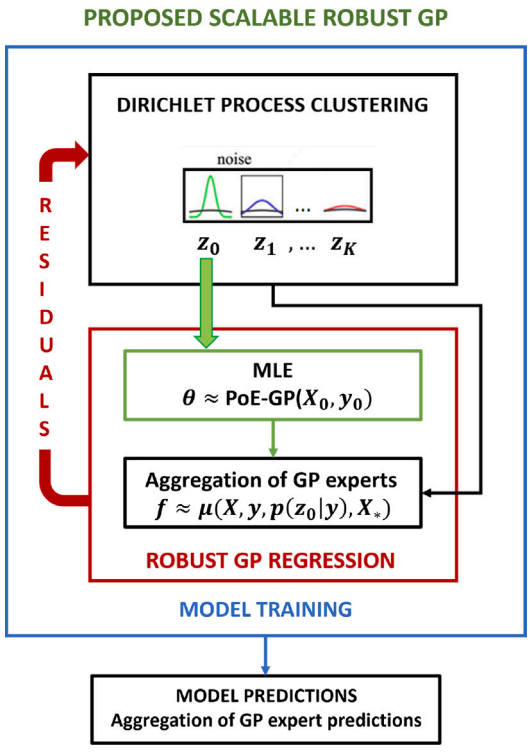


Fig. 2. Proposed scalable robust GP diagram.

here as G. A DP is defined by a concentration parameter α and a base distribution H:

$$G|\{H,\alpha\} \sim DP(H,\alpha)$$

As detailed in Blei and Jordan (2006b), we can characterise a DP as a stick-breaking prior:

$$\begin{aligned} \nu_k &\sim \text{Beta}(1,\alpha), & \pi_k &= \nu_k \prod_{j=1}^{k-1}(1-\nu_j) \\ \phi_k &\sim H, & G &= \sum_{k=1}^{\infty} \pi_k \delta_{\phi_k} \end{aligned} \tag{15}$$

Notice from Eq. (15) that the collection of discrete distributions $G(A_k)$ is defined over a measurable space Φ and indexed by partitions, A_k, such that, $A_k \subset \Phi$ and $\sum_{k=1}^{\infty} G(A_k) = G(\Phi) = 1$, describes a Dirichlet Process. DPs have been used to develop mixture models that provide clustering solutions in a non-parametric Bayesian fashion (Teh, 2010; Dilan, 2010; Blei and Jordan, 2006b). In the case of a DP, the infinite number of parameters that arises when $K \to \infty$ fulfils the non-parametric condition.

When describing mixture models with a DP, a countably infinite number of components are incorporated into the mixture. Using Eq. (15) to incorporate a countably infinite number of Gaussian distributions and writing Eq. (14) in terms of the precision $\tau_k = 1/\sigma_k^2$, the likelihood of $\boldsymbol{Z}$ and $\boldsymbol{\tau}$ is,

$$p(\boldsymbol{y}|\boldsymbol{Z},\boldsymbol{\tau}) = \prod_{n=1}^{N}\prod_{k=1}^{\infty} \mathcal{N}\left(r_n|0,\tau_k^{-1}\right)^{z_{nk}} \tag{16}$$

The parameters τ_k and z_{nk} are specified using a Gamma and stick-breaking prior (Eq. (15)), respectively, such that

$$\begin{aligned} \tau_k &\sim \text{Gamma}(c_0,d_0), & p(\boldsymbol{z}_n|\boldsymbol{\nu}) &= \prod_{k=1}^{\infty}\left[\nu_k^{z_{nk}}(1-\nu_k)^{z_{nj>k}}\right], \\ & & p(\boldsymbol{\nu}|\alpha) &= \prod_{k=1}^{\infty}\text{Beta}(\nu_k|1,\alpha) \end{aligned} \tag{17}$$

Defining $\boldsymbol{\Psi} = \{\boldsymbol{Z},\boldsymbol{\nu},\boldsymbol{\tau}\}$, the joint distribution of the observations and unknowns $\boldsymbol{\Psi}$ is,

$$p(\boldsymbol{y},\boldsymbol{\Psi}) = \prod_{k=1}^{\infty} p(\nu_k)p(\tau_k)\prod_{n=1}^{N} p(\boldsymbol{z}_n|\boldsymbol{\nu})p(y_n|\boldsymbol{z}_n,\boldsymbol{\tau}) \tag{18}$$

As the analytic solution of the posterior distribution $p(\boldsymbol{\Psi}|\boldsymbol{y})$ is intractable, we use a variational inference approximation (Blei and Jordan, 2006b). Let $q_\psi(\boldsymbol{\Psi})$ be a family of distributions (indexed by a variational parameter ψ) that represent an approximation of the true posterior $p(\boldsymbol{\Psi}|\boldsymbol{y})$. The logarithm of $p(\boldsymbol{y})$ can then be decomposed as follows,

$$\ln p(\boldsymbol{y}) = \underbrace{\int_{\boldsymbol{\Psi}} q_\psi(\boldsymbol{\Psi}) \ln \frac{p(\boldsymbol{y},\boldsymbol{\Psi})}{q_\psi(\boldsymbol{\Psi})} d\boldsymbol{\Psi}}_{\mathcal{L}(q)} - \underbrace{\int_{\boldsymbol{\Psi}} q_\psi(\boldsymbol{\Psi}) \ln \frac{p(\boldsymbol{\Psi}|\boldsymbol{y})}{q_\psi(\boldsymbol{\Psi})} d\boldsymbol{\Psi}}_{\text{KL}(p\|q)} \tag{19}$$

An approximation of the posterior $p(\boldsymbol{\Psi}|\boldsymbol{y})$ can be obtained either by minimising the KL divergence, $\text{KL}(p \parallel q)$, or by maximising the lower bound, $\mathcal{L}(q)$. As minimising $\text{KL}(p \parallel q)$ involves the unknown posterior $p(\boldsymbol{\Psi}|\boldsymbol{y})$, we choose to maximise $\mathcal{L}(q)$.

Any inference solution applied to the DP mixture model has to be computationally tractable, which implies that the infinite elements that form the random measure, $G = \sum_{k=1}^{\infty}\pi_k\delta_{\phi_k}$, must be truncated to a finite number. Consequently, for the variational approach, the number of components is fixed to a value T. To form the variational approximation, we now choose to restrict our search to variational distributions $q(\boldsymbol{\Psi})$ that factorise as follows,

$$q(\boldsymbol{\Psi}) = \prod_{k=1}^{T-1} q_{\alpha_k}(\nu_k)\prod_{k=1}^{T} q_{\gamma_k}(\tau_k)\prod_{n=1}^{N} q_{\zeta_n}(\boldsymbol{z}_n) \tag{20}$$

were $q_{\alpha_k}(\nu_k)$ are Beta distributions indexed by parameters α_k, $q_{\gamma_k}(\tau_k)$ are exponential family distributions indexed by parameters γ_k, and $q_{\zeta_n}(\boldsymbol{z}_n)$ are multinomial distributions indexed by parameters ζ_n. Inference based on the factorised form shown in Eq. (20) is called *mean field variational inference* (Bishop, 2006) and, in the present application, instead of treating T as a variational parameter, we assume T is the upper limit of the number of components (Blei and Jordan, 2006b).

Substituting Eq. (20) into the lower bound defined in Eq. (19), the logarithm of the resulting optimal distributions can be shown to be

$$\ln q^*(\boldsymbol{\omega}) = \mathbb{E}_{\boldsymbol{\Psi}\setminus\{\boldsymbol{\omega}\}}\left[\ln p(\boldsymbol{y},\boldsymbol{Z},\boldsymbol{\nu},\boldsymbol{\tau})\right] + \text{const} \tag{21}$$

$$\ln q^*(\boldsymbol{Z}) = \sum_{n=1}^{N}\sum_{k=1}^{T} z_{nk}\ln\left(\frac{\rho_{nk}}{\sum_{i=1}^{T}\rho_{ni}}\right) \tag{22}$$

$$\ln q^*(\boldsymbol{\nu}) = \ln\sum_{k=1}^{T}\left(\frac{\Gamma(a_k+b_k)}{\Gamma(a_k)\Gamma(b_k)}\nu_k^{a_k-1}(1-\nu_k)^{b_k-1}\right) \tag{23}$$

$$\ln q^*(\boldsymbol{\tau}) = \ln\left(\sum_{k=1}^{T}\frac{1}{\Gamma(c_k)}d_k^{c_k}\tau_k^{c_k-1}\exp\{-d_k\tau_k\}\right) \tag{24}$$

where $\mathbb{E}_{\boldsymbol{\Psi}\setminus\{\boldsymbol{\omega}\}}$ stands for the expectation with respect to all the elements of $\boldsymbol{\Psi}$ except for $\boldsymbol{\omega}$ and $\ln\rho_{nk} = \mathbb{E}[\ln\nu_k] + \sum_{j=1}^{k-1}\mathbb{E}[\ln(1-\nu_j)] + \frac{1}{2}\left(\mathbb{E}[\ln\tau_k] - \mathbb{E}[r_n^2\tau_k] + d\ln 2\pi\right)$. In Eq. (23) the choice of ν_k is governed by a Beta distribution with parameters, $a_k = 1 + \sum_{n=1}^{N} q(z_{nk}|r_n)$, and $b_k = \alpha + \sum_{n=1}^{N} q(z_{n,j>k}|r_n)$, where α is the parameter of the Beta prior in Eq. (17). Furthermore, from Eq. (24) we see that the choice of τ_k is governed by a Gamma distribution with parameters, $c_k = c_0 + \frac{1}{2}\sum_{n=1}^{N} q(z_{nk}|r_n)$, and $d_k = d_0 - \frac{1}{2}\sum_{n=1}^{N} q(z_{nk}|r_n)r_n^2$, where c_0 and d_0 are the parameters of the Gamma prior in Eq. (17).

4.2. The regression step

In the regression step, we take advantage of the information provided by the clustering step and exploit the closed-form expressions associated with standard GP regression. Using only the observations corrupted with noise from the Gaussian whose standard deviation is

$\sigma_0 = \min\{\sigma_1, \ldots, \sigma_K\}$, we can return to a standard GP formulation and use Eq. (8) to define the log-likelihood as

$$\ln p(\boldsymbol{y}_0|\,\boldsymbol{\theta}) = -\frac{1}{2}\ln|\boldsymbol{C}_0| - \frac{1}{2}\boldsymbol{y}_0^T\boldsymbol{C}_0^{-1}\boldsymbol{y}_0 - \frac{N_0}{2}\ln(2\pi) \tag{25}$$

where the training dataset, the number of observations, and the ijth element of $\boldsymbol{C}_0$ in Eq. (25) are, $\mathcal{D}_0 = \{\boldsymbol{X}_0, \boldsymbol{y}_0\}$, N_0, and $k(\mathbf{x}_{i,0}, \mathbf{x}_{j,0}) + \delta_{ij}\sigma_0^2$, respectively. We can now apply a MLE procedure to Eq. (25) to estimate the GP hyperparameters.

We now make use of the estimated allocation variables to give an approximation of $\boldsymbol{f}$. The OMGP assumes that there exists J different latent functions $\{\boldsymbol{f}_j\}_{j=1}^{J}$ (called trajectories) that are associated with J sets of observations; these observations can, for example, represent the trajectory of moving sources (missiles, aircraft, etc.) The OMGP predictive distribution for the jth trajectory at a new input $\mathbf{x}_*$ is given by, $\mathcal{N}\left(f_{j*}|\mu_j(\mathbf{x}_*), \sigma_j(\mathbf{x}_*)^2\right)$, where $\mu_j(\mathbf{x}_*) = \boldsymbol{k}_*^T(\boldsymbol{K} + \boldsymbol{R}_j^{-1})^{-1}\boldsymbol{y}$ and $\sigma_j(\mathbf{x}_*)^2 = \sigma^2 + k_{**} - \boldsymbol{k}_*^T(\boldsymbol{K} + \boldsymbol{R}_j^{-1})^{-1}\boldsymbol{k}_*$; where,

$$\boldsymbol{R}_j = \begin{pmatrix} \tilde{r}_{n=1,j} & 0 & \cdots & 0 \\ 0 & \tilde{r}_{n=2,j} & 0 & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{r}_{n=N,j} \end{pmatrix}$$

where

$$\tilde{r}_{nk} = \frac{\rho_{nk}}{\sum_{i=1}^{T}\rho_{ni}} \tag{26}$$

is often defined as the *responsibility* of the Gaussian associated with $z_{nk} = 1$ for generating observation y_n (Murphy, 2012; Bishop, 2006).

If we associate the jth trajectory with the noise source identified as having variance σ_0^2, we can use the OMGP predictive mean and variance to realise estimates of $\boldsymbol{f}$ and the associated uncertainty by calculating,

$$\mu(\mathbf{x}_*) = \boldsymbol{k}_*^T(\boldsymbol{K} + \boldsymbol{R}_0^{-1})^{-1}\boldsymbol{y} \tag{27}$$

$$\sigma(\mathbf{x}_*)^2 = \sigma^2 + k_{**} - \boldsymbol{k}_*^T(\boldsymbol{K} + \boldsymbol{R}_0^{-1})^{-1}\boldsymbol{k}_* \tag{28}$$

where the vector $\boldsymbol{k}_*$ has elements $k(\mathbf{x}, \mathbf{x}_*)$ and $k_{**} = k(\mathbf{x}_*, \mathbf{x}_*)$. Notice that the only difference between the standard GP distribution (Bishop, 2006) and Eqs. (27) and (28) is the addition of $\boldsymbol{R}_0$ to the kernel matrix $\boldsymbol{K}$. Specifically, as the posterior probability that an observation y_n belongs to the cluster $k = 0$ (responsibility $\tilde{r}_{n0}$) decays, the amount of noise associated with that observation increases proportionally, thus reducing its effect on the predictive mean and variance.

Thus far we have described the implementation of a single Gaussian Process that, though the application of a Dirichlet Process clustering step, employs a mixture of Gaussian likelihood; we refer to this algorithm as the 'DPGP'. Regarding scalability to large datasets, we use a product-of-expert approach to distribute the computations involved in the regression step. By partitioning the training dataset into M subsets we can assign a GP expert to each of the M partitions and use the gPoE scheme to aggregate the experts predictions on the complete dataset of size N. The resulting approach - referred to here as the Distributed DPGP (DDPGP) - provides a mechanism for the management of data lifecycles (criterion 2 in Section 1; modularity), as individual DPGP experts need only be omitted from the product-of-expert predictive calculations if the relevance of the data used to train those experts is judged to have reduced.

It is important to note that the proposed approach is different from Gaussian Mixture Regression (and its variants e.g. Variational Bayesian Gaussian Mixture Regression Zhu et al., 2016), though the terminology is similar. With Gaussian Mixture Regression, both the model's inputs and outputs ($\boldsymbol{x}$ and y in the current notation) are treated as random variables before a Gaussian Mixture Model is used to model the joint distribution $p(\boldsymbol{x}, y)$; subsequent predictions are then computed from the conditional distribution, $p(y|\boldsymbol{x})$. The proposed approach, in contrast, models the latent function (denoted f in the current notation) as a Gaussian Process, before then assuming that the observations of the latent function have been corrupted by noise drawn from a Gaussian Mixture Model. Our approach also differs from that presented in Yu (2012) (again, despite similar terminology). In Yu (2012) (which uses a 'finite mixture model based Gaussian Process regression approach'), input data is first clustered using a Gaussian Mixture Model before separate Gaussian Processes are trained on data from each of the identified clusters (the aim being that separate Gaussian Processes now represent different operating modes of the process). Each GP, however, utilises a standard Gaussian likelihood and, therefore, is not robust to outliers relative to the approach proposed in the current paper.

4.3. Satisfaction of criteria

In Section 1, 5 key criteria that the proposed solution must satisfy, established from work within the continuous manufacturing space, were described. With the technical details of our approach established, we now summarise how the proposed solution satisfy these 5 criteria:

1. Scalability. By using a product-of-experts approach, adding information from additional data simply involves training a new GP expert model which can then be included in subsequent predictions. The computational cost of adding additional data is therefore independent of the size of previous training data, allowing scalability to large datasets.
2. Modularity. If it is decided that previous data is no longer relevant to current operation, the expert trained on that data can simply be removed from subsequent predictions. This feature of the approach facilitates management of the data lifecycle.
3. Stable out-of-data performance. Being a Gaussian Process approach, it is known that the model will converge to its prior statistics when applied far from the training data.
4. Uncertainty quantification. Being a Gaussian Process approach, closed-form expressions for the predictive standard deviation can be used to facilitate uncertainty quantification.
5. Robustness to unrepresentative data. By using a Gaussian Mixture likelihood, we are able to identify and subsequently ignore outlier data.

5. Experiments using synthetic datasets

The current section details comparisons between the proposed approach, a standard GP and Stegle's robust GP (RGP) (Stegle et al., 2008). We compare with the RGP as it satisfies three out of the five criteria (Stable out-of-data, Uncertainty quantification, and Robustness to representative data). We note, however, that the RGP does not scale well to large datasets and, as a result, the analysis described in the current section uses a relatively small synthetic dataset to avoid memory issues. Comparisons between the gPoE and proposed DDPGP are shown in Section 6, where we use a number of data points that can be problematic for the standard GP and RGP approaches.

We use a synthetic dataset to assess the suitability of a standard GP, RGP and the proposed DPGP when using data corrupted with noise sampled from a mixture of Gaussian distributions. Specifically, $N = 150$ realisations of the function, $f(x) = 150x\sin(x)$, were corrupted following the noise model described in Eq. (11). Knowing the function from which the observations are generated, the models' predictive accuracy is evaluated in terms of the Root Mean Squared Error (RMSE).

The mixture of Gaussian distributions was created using $K = 3$ independent components with the following parameters:

- Proportionalities: $\pi_0 = 0.5, \pi_1 = 0.4, \pi_2 = 0.1$
- Standard Deviations: $\sigma_0 = 10, \sigma_1 = 90, \sigma_2 = 300$

Using the Squared Exponential Kernel (Eq. (3)), the initial estimates of the kernel parameters for the standard GP, RGP and DPGP models were set to $\sigma_f = 1, \ell = 1,$ and $\sigma = 0.5$. Notice that we cannot initialise the same mixture parameters for the RGP and DPGP as the number of

Table 1
RMSE of 300 predictions at inputs not used for training. Please, see the text for a detailed explanation of Approach 1 and 2.

Root mean square error			
Obtained with Approach 1		Obtained with Approach 2	
GP	1647.42	GP	1647.42
RGP	69.96	RGP	26.80
DPGP	25.10	DPGP	21.58

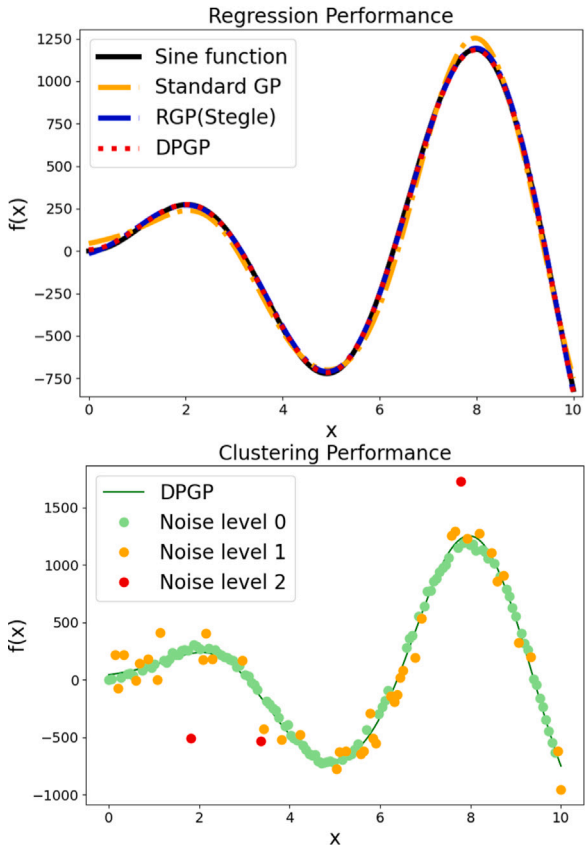


Fig. 3. *Top:* A comparison of the standard GP, RGP and DPGP predictive mean after the 3 models have been trained with the synthetic data. *Bottom:* The clustering outcome of the DPGP where different colours have been used to identify each observation with a Gaussian component from the mixture of Gaussian distributions.

mixture parameters for the DPGP depends on the upper bound, T. To initialise such models with the same mixture parameters and provide a fair comparison of their calculated RMSE, we take two approaches.

Approach 1: it is assumed that the number of mixture components is known and set $K = 3$ and $T = 3$ for the RGP and the DPGP, respectively. The same initial mixture parameters can now be chosen (randomly) for both models. After training, their predictive accuracy is measured in terms of the RMSE at 300 test points, as shown in Table 1 (left).

Approach 2: the initial mixture parameters are chosen randomly 100 times for the RGP with $K = 3$ and DPGP with $T \in \{6, .., 10\}$. Subsequently, the lowest RMSE values for the RGP and DPGP were chosen for accuracy comparison, as shown in Table 1 (right).

A visual comparison of the models' predictive mean is shown in Fig. 3 (top), whereas the DPGP clustering outcome is shown in Fig. 3 (bottom). Plots in Fig. 3 were obtained using Approach 2. We note that the DPGP correctly identified the number of mixture components to be $K = 3$.

The visual comparison in Fig. 3 and the quantitative comparison in Table 1 show that the RGP and DPGP successfully ignored the observations corrupted with Gaussian distributions whose $\sigma^2 > \sigma_0^2$ when learning the latent function from the corrupted data. In addition, Fig. 3 illustrates that the standard GP predictive mean deviates from the sine function at outlier positions $x = 0, x = 2$ and $x = 8$. Accordingly, the standard GP RMSE is the highest. The quantitative comparisons in Table 1 showed that the DPGP model was the most accurate in estimating the sine function using the synthetic data.

6. Case study

The data used in the present case study was provided by a company from one of the UK's foundation industries. The model's inputs correspond to readings from 27 sensors, each taking measurements at different stages of the manufacturing process. Fault density, observed at the end of the process, could be the result of changes in the manufacturing process that occurred between several hours or several days beforehand. In the current work the amount of time between the observation of a phenomenon in the manufacturing process and its subsequent effect on product quality is referred to as a 'time lag'.

The authors note that, as a result of the commercial sensitivities, specific values of fault density and some specifics regarding the model development process are not reported in the following. We believe, however, that the visual analysis afforded by the figures reported in this paper sufficiently demonstrate the advantages of the proposed approach and justify the conclusions described in Section 8.

6.1. Data exploration and pre-processing

Firstly, inputs were removed which, for instance, were found to be very low-resolution or constant over the time duration of interest. This left 22 inputs remaining. Missing values for each input were then replaced using linear interpolation before each signal was passed through a low-pass filter to remove high frequency noise. All signals were standardised to be zero-mean and unit-variance before estimation of the time lags associated with each input. Time lags were initially identified using process knowledge provided by the project's industrial partner. These estimates were later refined using a random sampling approach whereby multiple training runs were conducted using time lag samples that were drawn from probability distributions centred on the initial estimates.

6.2. Model training

The training data consisted of $N = 17{,}000$ input–output pairs which were split into $M = 17$ adjacent regions, such that each expert was assigned 1000 training points. We note that the size of the training set was dictated by data-availability, rather than the scalability of the proposed approach. The initial mixture parameters of each DPGP expert were chosen randomly with the upper bound of the number of mixture components fixed at $T = 7$.

The hyperparameters inferred when training each DPGP expert provides information regarding the relevance of the inputs over each of the M regions (recall from Eq. (5) that the model sensitivity is high for the inputs whose length scale values are low). Accordingly, once the DDPGP was trained, the inputs with lowest influence were remvoed, reducing the number of inputs down to 10. Model training was then repeated using these 10 inputs. We emphasise that, for this case study, each DPGP expert was allowed to have different hyperparameters (though it is also possible to constrain the hyperparameters of each expert to be the same); this allows us a degree of flexibility over the approach described in Zhou et al. (2021), where a single set of hyperparameters is used to describe the entire training set. We also note that, while data was partitioned by time period in the current case study, other data partition strategies could also be used; the data assigned to each expert could be dictated by product type or using a randomised sampling approach, for example.

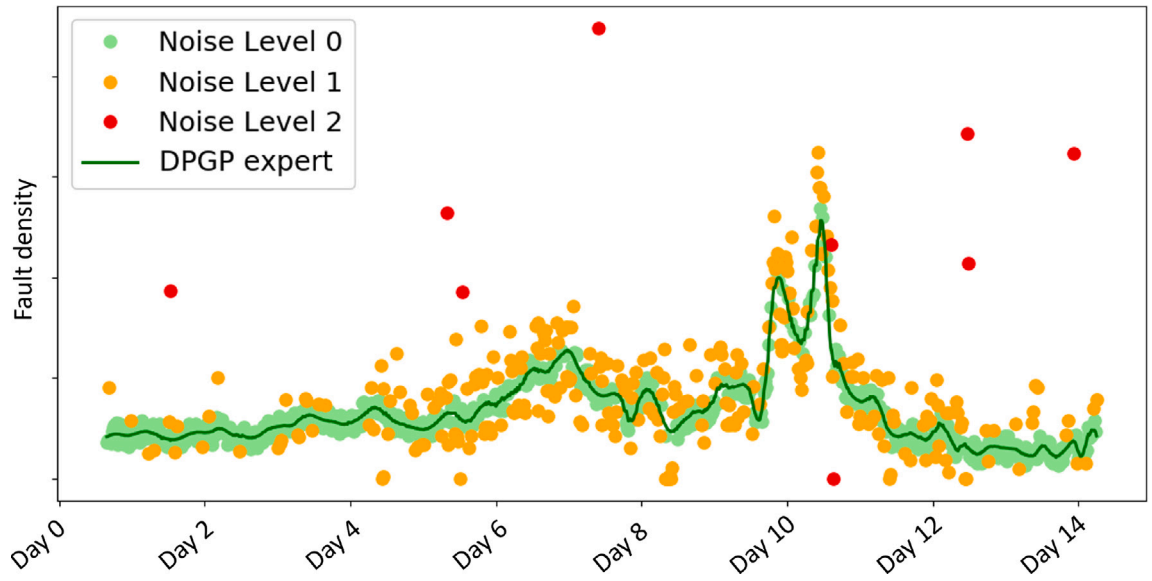


Fig. 4. Example DDPGP clustering results. The (normalised) number of observed data points have been assigned to one of the $K = 3$ Gaussian clusters with different colours representing the different values of noise variance.

Table 2
The colours in each row have been used to identified an observation with each Gaussian cluster in the mixture. Columns two and three show the DPGP estimated mixture parameters, π and σ, respectively.

Estimates of the mixture parameters		
Colour	π	σ (Faults/10 m^2)
Green	0.45	0.7
Yellow	0.36	0.25
Red	0.18	1.5

6.3. Results

In this section we analyse the model's ability to identify and remove outliers in the product quality data as well as the model's ability to provide future predictions of fault density (and associated uncertainties).

The DPGP experts identified a minimum of $K = 3$ and a maximum of $K = 5$ clusters at the $M = 17$ different regions of training data. In the following, a period where a rise in the number of faults was known to have occurred is used to analyse the model's clustering performance. The clustering outcome for the DPGP expert trained over such a period is shown in Fig. 4 where it can be seen that the DPGP expert inferred the presence of $K = 3$ clusters; we have associated each Gaussian cluster with a colour shown in the first column of Table 2. The mixture parameters estimated by the DPGP expert are shown in Table 2.

The observations that were used to infer the model (the green points) are, we hypothesise, generated by measured fluctuations in the manufacturing process while the remaining observations are assumed to have been generated by external processes. Two fault increases associated with the manufacturing process were identified; the first one is a slow increase whose peak is between Day 6 and Day 8, while the second illustrates a relatively fast increase whose peak is between Day 10 and Day 11. Discussion with the project's industrial revealed that these rises were indeed due to fluctuations in the manufacturing process that were measured and used as inputs to the model, while the data associated with the red points were either erroneous sensor readings or caused by production process that cannot be captured by the model (e.g. repairs or maintenance).

Comparisons between the gPoE and DDPGP are shown in Fig. 5. Specifically, for the same time period, a comparison of the predictive mean results between a single GP expert (yellow line), corresponding to the gPoE model, and a DPGP expert (red line), corresponding to the DDPGP model, is shown in Fig. 5. Notice that the predictive mean of the standard GP expert is affected by the spikes in fault density that occur on Days 7, 12 and 14. It can also be seen that, the 3σ GP expert confidence bounds (Fig. 5 top) are more conservative than the confidence bounds associated with the DPGP predictions (pink in Fig. 5 bottom).

We now focus on a second region, where the presence of outliers is more obvious. Fig. 6 shows the predictive means of a single GP and DPGP expert; this time the GP is overfitting, following almost exactly all of the fault density observations (including those that are, by eye, clearly outliers) while, by ignoring outliers, the DPGP has not overfit the data to same extent. Both models were trained using the same initial GP hyperparameters.

Given that we require the model to predict future increases in the number of product faults, the predictive performance of the DDPGP was then evaluated using data that was not used in training. Specifically, after the model was trained, its performance was evaluated on 14-days of test data. Fig. 7 shows the gPoE and DDPGP predictions at the last 1000 training points on the left-hand side of the vertical dashed line. The gPoE and DDPGP predictions on 1000 test points corresponding to 14-days of 'unseen' data are shown on the right hand-side of the vertical dashed line. Notice that, even though the gPoE does not (at least visually) look to have been overfit to the training data, it struggles to follow the measured fault density over the testing period. Turning our attention on the DDPGP predictions, the predictive mean more closely follows the slow-varying trends in the measurements of fault density, capturing an increase in the number of faults between Days 9 and 10. Post-analysis by the project's industrial partner that, again, the rises predicted by the model where due to measured fluctuations in the industrial process.

7. Future work

Throughout this work, the hyperparameters of the regression model are tuned using a maximum-likelihood approach. We note that a more comprehensive approach would seek to quantify (and propagate) the uncertainties associated with these hyperparameters. This is typically achieved by sampling from the hyperparameter posterior distribution using, for example, Markov Chain Monte Carlo or Sequential Monte Carlo samplers (Del Moral et al., 2006) though the computational cost of such an approach may be prohibitive for large sets of training data.

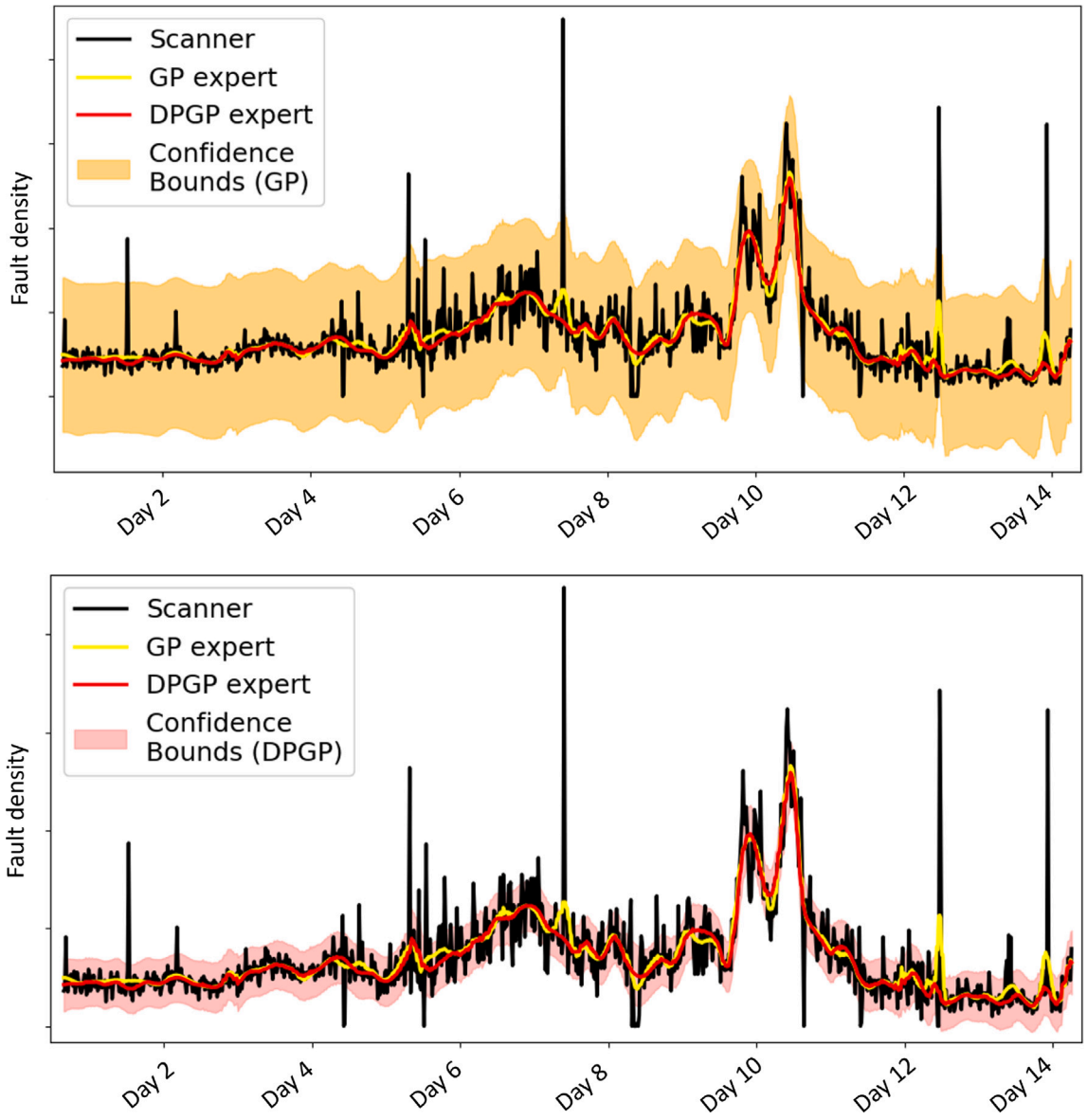


Fig. 5. The predictive means of the standard GP (yellow line) and DPGP (red line). *Top:* The orange shaded area corresponds to $\pm 3\sigma^{GP}$ from the GP predictive mean. *Bottom:* The pink shaded area corresponds to $\pm 3\sigma_0$ from the DPGP predictive mean.

Regarding the use of a mixture of Gaussian noise model we note that, while the non-parametric nature of the likelihood makes it very flexible, it still assumes that the statistics of the underlying noise model are stationary; though this has been found to be appropriate for the industrial case study described in Section 6, it may not always be true. Moreover, as is common with many machine learning approaches, the final performance of the model will be dependent to some degree on the initial conditions of the optimisation routine (though this can be addressed to some extent using established approaches e.g. randomising initial conditions and selecting the optimal results using k-fold validation).

In the industrial example described in Section 6, allowing each GP expert to have different hyperparameters was found to be beneficial. We note, however, that this approach can also lead to a form of overfitting where each expert is overly 'tuned' to its own training data. This effect may be mitigated by, for example, increasing the amount of training data received by each expert or constraining experts to have the same hyperparameters.

Our approach scales well to large data sets, as the incorporation of additional data simply involves training an additional GP expert. When predictions are required, the sequential loading of each expert into memory also allows us to avoid memory overflow issues. We note, however, that such an approach limits the real-time capability of the predictive process, as the sequential loading of many GP models can be time consuming. Improving real-time performance is a topic of future work, though a possible strategy for addressing this could involve utilising faster Sparse GPs (e.g. Titsias (2009)) in place of 'full' GPs for each expert.

The current paper describes the construction of a predictive model that can then be used to optimise a continuous manufacturing process. We do not, however, describe how this optimisation may be conducted; for more information in this regard Refs. Tresp (2001), Titsias (2009), Deisenroth and Wei Ng (2015) and Liu et al. (2018a) describe suitable approaches from the context of 'Robust Parameter Design'.

8. Conclusions

This paper describes the development of an 'AI' (i.e. data-based) model which, when applied to an application in one of the UK's foundation industries, can be used to realise future predictions of product

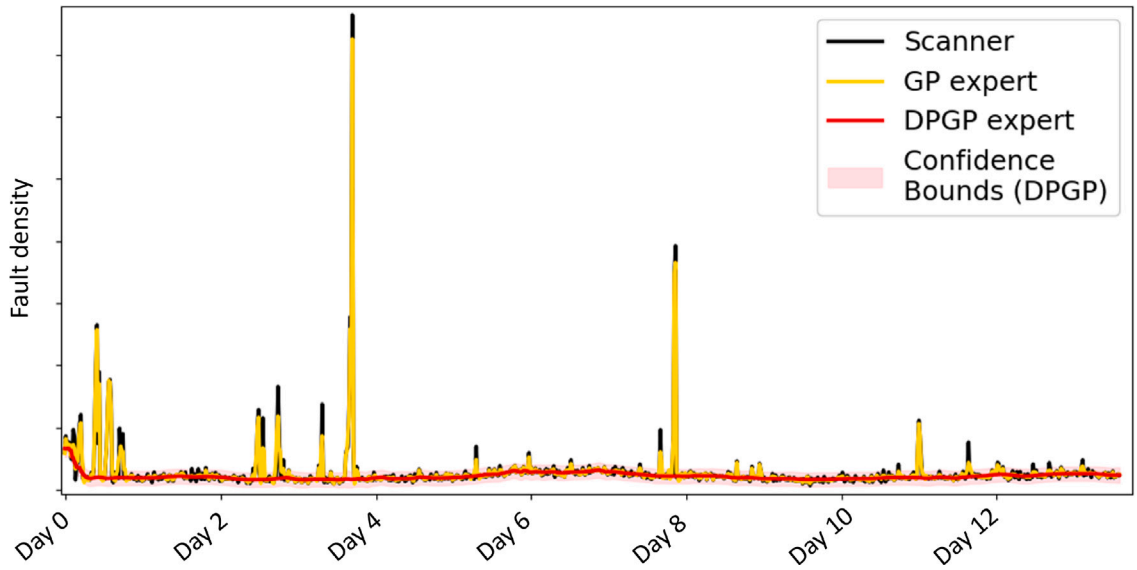


Fig. 6. The predictive means of the standard GP (yellow line) and DPGP (red line) $m = 11$ experts. The pink shaded area corresponds to $\pm 3\sigma_0$ from the DPGP predictive mean.

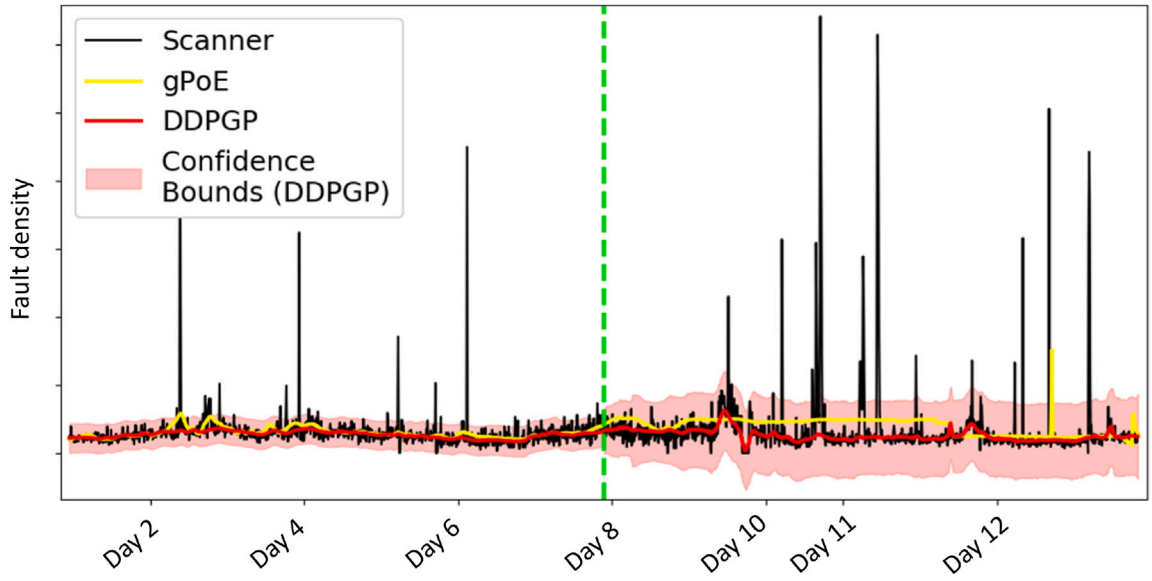


Fig. 7. The predictive means of the gPoE and DDPGP on training (left-hand side of the vertical dashed line) and testing (right-hand side of the vertical dashed line) data.

quality. Based on the author's experience working within this industrial setting, 5 key criteria are described which, we believe, must be considered from the very start of the model development process: scalability, modularity, stable out-of-data performance, uncertainty quantification and robustness to unrepresentative data. To that end, we propose and demonstrate a Gaussian Process regressor whose noise model is defined as a Dirichlet Process, before also adopting a product-of-experts model to ensure scalability and modularity in our approach. As well as demonstrating that the model satisfies the 5 key criteria, results from the industrial case study provide evidence that the proposed model provides better predictions of product quality than a standard Gaussian Process with Gaussian noise model.

CRediT authorship contribution statement

Diego Echeverria-Rios: Methodology, Software, Investigation, Data curation, Writing. **Peter L. Green:** Conceptualization, Methodology, Writing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data that has been used is confidential.

References

Alcorn, M.A., Li, Q., Gong, Z., Wang, C., Mai, L., Ku, W.-S., Nguyen, A., 2019. Strike (with) a pose: Neural networks are easily fooled by strange poses of familiar objects. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4845–4854.

Arora, J.S., 2006. Jan a. Snyman, practical mathematical optimization: An introduction to basic optimization theory and classical and new gradient-based algorithms. Struct. Multidiscip. Optim..

Bar-Shalom, Y., Fortmann, T.E., Cable, P.G., 1990. Tracking and data association. J. Acoust. Soc. Am..

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.
Blei, D.M., Jordan, M.I., 2006a. Variational inference for Dirichlet process mixtures. Bayesian Anal..
Blei, D.M., Jordan, M.I., 2006b. Variational inference for Dirichlet process mixtures. Bayesian Anal. 1 (1), 121–144.
Chati, Y.S., Balakrishnan, H., 2017. A Gaussian process regression approach to model aircraa engine fuel flow rate. In: 2017 ACM/IEEE 8th International Conference on Cyber-Physical Systems (ICCPS), Vol. 10.
Cox, I.J., 1993. A review of statistical data association techniques for motion correspondence. Int. J. Comput. Vis. 10 (1), 53–66.
Deisenroth, M.P., Fox, D., Rasmussen, C.E., 2015. Gaussian processes for data-efficient learning in robotics and control. IEEE Trans. Pattern Anal. Mach. Intell. 37 (2), 408–423, [Online]. Available: http://www.ieee.org/publications_standards/publications/rights/index.html.
Deisenroth, M.P., Wei Ng, J., 2015. Distributed Gaussian processes. In: Proceedings of the 32nd International Conference on Machine Learning. Proceedings, Lille, [Online]. Available: http://proceedings.mlr.press/v37/deisenroth15.pdf.
Del Moral, P., Doucet, A., Jasra, A., 2006. Sequential monte carlo samplers. J. R. Stat. Soc. Ser. B Stat. Methodol. 68 (3), 411–436.
Dilan, G., 2010. Dirichlet process Gaussian mixture models: Choice of the base distribution. J. Comput. Sci. Tech. 25 (July), 615–626.
Duvenaud, D.K., 2014. WI-EG-029 (Paraquat) (Ph.D. dissertation). University of Cambridge.
Feng, Z., Li, D., Qin, G., Liu, S., 2009. Effect of the flow pattern in a float glass furnace on glass quality: Calculations and experimental evaluation of on-site samples. J. Am. Ceram. Soc. 92 (12), 3098–3100.
Feng, Z., Wang, J., Ma, Y., Tu, Y., 2021. Robust parameter design based on Gaussian process with model uncertainty. Int. J. Prod. Res. 59 (9), 2772–2788.
Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E., 1991. Adaptive mixtures of local experts.
Jin, D., Xu, S., Tong, L., Wu, L., Liu, S., 2020. A deep learning model for striae identification in end images of float glass. Trait. Signal 37 (1), 85–93.
Jylänki, P., Vanhatalo, J., Vehtari, A., 2011. Robust gaussian process regression with a student-t likelihood. J. Mach. Learn. Res. 12, 3227–3257.
Kuss, M., 2006. Gaussian Process Models for Robust Regression , Classification , and Reinforcement Learning (Ph.D. Thesis).
Lázaro-Gredilla, M., Titsias, M.K., 2011. Variational heteroscedastic Gaussian process regression. In: 28th International Conference on Machine Learning (ICML-11. pp. 841–848, [Online]. Available: http://eprints.pascal-network.org/archive/00009089/.
Lázaro-Gredilla, M., Van Vaerenbergh, S., Lawrence, N.D., 2011. Overlapping Mixtures of Gaussian Processes for the data association problem. Pattern Recognit. 45, 1386–1395, [Online]. Available: www.elsevier.com/locate/pr.
Liu, H., Cai, J., Ong, Y.S., Wang, Y., 2019. Understanding and comparing scalable Gaussian process regression for big data. Knowl.-Based Syst. 164, 324–335.
Liu, H., Cai, J., Wang, Y., Ong, Y.S., 2018a. Generalized robust Bayesian committee machine for large-scale Gaussian process regression. In: 35th International Conference on Machine Learning, ICML 2018, Vol. 7. pp. 4898–4910.
Liu, H., Ong, Y.-S., Cai, J., 2018b. Large-scale heteroscedastic regression via Gaussian process. J. Wuhan Univ. Technol. 1–14, [Online]. Available: http://arxiv.org/abs/1811.01179.
Marcus, G., 2020. The next decade in AI: four steps towards robust artificial intelligence. arXiv preprint arXiv:2002.06177.
Murphy, K.P., 2012. Machine Learning: A Probabilistic Perspective. MIT Press.
Neal, R.M., 1997. Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. Tech. Rep. 9702, University of Toronto, Toronto, pp. 1–24.
Qin, Y., Lou, Z., Wang, Y., Lu, S., Sun, P., 2022. An analytical partial least squares method for process monitoring. Control Eng. Pract. 124, 105182.
Rasmussen, C.E., 2006. Gaussian Processes for Machine Learning. The MIT Press.
Ross, J.C., Dy, J.G., 2013. Nonparametric mixture of Gaussian processes with constraints. In: W, J., CP (Eds.), 30th International Conference on Machine Learning. Atlanta, Georgia, [Online]. Available: http://proceedings.mlr.press/v28/ross13a.pdf.
Stegle, O., Fallert, S.V., Mackay, D.J.C., Brage, S., 2008. Gaussian process robust regression noisy heart rate data. Engineering 55 (9), 2143–2151.
Sun, S., Xu, X., 2011. Variational inference for infinite mixtures of Gaussian processes with applications to traffic flow prediction. IEEE Trans. Intell. Transp. Syst. 12 (2), 466–475.
Teh, Y.W., 2010. A Dirichlet process mixture of hidden Markov models for protein structure prediction. Ann. Appl. Stat. 4 (2), 916–942.
Titsias, M.K., 2009. Variational learning of inducing variables in sparse Gaussian processes. In: van Dyk, D., Welling, M. (Eds.), Proceedings of the Twelfth International Workshop on Artificial Intelligence and Statistics. JMLR W&CP, Clearwater Beach, FL, pp. 567–574, [Online]. Available: http://proceedings.mlr.press/v5/titsias09a/titsias09a.pdf.
Tresp, V., 2001. Mixtures of Gaussian processes. Adv. Neural Inf. Process. Syst. 13, 654–660.
Vanhatalo, J., Jylanki, P., Vehtari, A., 2009. Gaussian process regression with Student-t likelihood. In: Advances in Neural Information Processing Systems 22 - Proceedings of the 2009 Conference. pp. 1910–1918.
Wang, K., Pleiss, G., Gardner, J., Tyree, S., Weinberger, K.Q., Wilson, A.G., 2019. Exact Gaussian processes on a million data points. In: Advances in Neural Information Processing Systems, Vol. 32.
Worden, K., Green, P.L., 2016. A machine learning approach to nonlinear modal analysis. Mech. Syst. Signal Process. 84, 34–53, [Online]. Available: http://dx.doi.org/10.1016/j.ymssp.2016.04.029i.
Yu, J., 2012. Online quality prediction of nonlinear and non-Gaussian chemical processes with shifting dynamics using finite mixture model based Gaussian process regression approach. Chem. Eng. Sci. 82, 22–30.
Zhou, X., Gao, Y., Jiang, T., Feng, Z., 2023. An online approach for robust parameter design with incremental Gaussian process. Qual. Eng. 35 (3), 430–443.
Zhou, X., Jiang, T., Zhou, Z., Hu, X., 2021. Sequential-support vector regression based online robust parameter design. Comput. Ind. Eng. 158, 107391.
Zhu, J., Ge, Z., Song, Z., 2016. Variational Bayesian Gaussian mixture regression for soft sensing key variables in non-Gaussian industrial processes. IEEE Trans. Control Syst. Technol. 25 (3), 1092–1099.
Zhu, J., Ge, Z., Song, Z., Gao, F., 2018. Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data. Annu. Rev. Control 46, 107–133, [Online]. Available: https://doi.org/10.1016/j.arcontrol.2018.09.003.