Article:
Tsialiamanis, G. orcid.org/0000-0002-1205-4175, Sbarufatti, C. orcid.org/0000-0001-5511-8194, Dervilis, N. orcid.org/0000-0002-5712-7323 et al. (1 more author) (2024) On a meta-learning population-based approach to damage prognosis. Mechanical Systems and Signal Processing, 209. 111119. ISSN 0888-3270

https://doi.org/10.1016/j.ymssp.2024.111119

Mechanical Systems and Signal Processing 209 (2024) 111119

Contents lists available at ScienceDirect

Mechanical Systems and Signal Processing

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

On a meta-learning population-based approach to damage prognosis

G. Tsialiamanis [a,*], C. Sbarufatti [b], N. Dervilis [a], K. Worden [a]

[a] Dynamics Research Group, Department of Mechanical Engineering, University of Sheffield, Mappin Street, Sheffield S1 3JD, United Kingdom
[b] Politecnico di Milano, Department of Mechanical Engineering, Via La Masa 1, 20156 Milano, Italy

ARTICLE INFO

Communicated by E. Chatzi

Keywords:
Population-based structural health monitoring (PBSHM)
Damage prognosis
Machine learning
Meta-learning

ABSTRACT

The current work studies the application of population-based structural health monitoring (PBSHM) to the problem of damage prognosis. Two methods are proposed for population-informed damage prognosis and they are evaluated according to their performance using an experimental dataset. The first method is an attempt to define a functional subspace, which includes the potential behaviour of members of the population subjected to the phenomenon of damage evolution. The second approach is a *meta-learning* method, the *deep kernel transfer* (DKT) method, which seeks to exploit information from a population in order to enhance the predictive performance of a *Gaussian process*. The predictive capabilities of the two methods are tested in an experimental crack-growth problem. The results reveal that the two methods are properly informed by the population to make predictions about new structures and show potential in dealing with the problem of damage evolution, which is a problem of imbalanced and difficult-to-acquire data.

1. Introduction

This paper is a continuation of a sequence devoted to introducing foundations for a new discipline of Population-based Structural Health Monitoring (PBSHM) [1–5]. The new technology is an attempt to deal with a quite common problem of *structural health monitoring* (SHM) [6] - the scarcity of data. The various modelling procedures of structural health monitoring are indivisibly connected to the acquisition and the exploitation of data. As a result, a convenient and powerful approach to follow is to define *data-driven* models to perform the desired monitoring and modelling of the structures [7–9]. However, for such approaches to be effective, data should be available from the various problems and phenomena, which one seeks to model.

In many cases, such data are not available for structures, either because they have not been appropriately monitored or because they have only been recently erected. In other cases, structures may be extensively monitored and one may have access to data instances from different environmental or operational conditions from these structures, as well as access to data from damaged states, a vital element needed for the definition of various types of data-driven SHM models. At the same time, it is clear that structures are made of similar materials, operate in similar environmental conditions, and have similar shapes and connectivity. In many cases structures are even nominally identical, i.e. they are manufactured as identical and their differences could be apportioned to inherently random events. Therefore, one could consider populations of structures, which share similar physics and, consequently, knowledge becomes transferable between them. This lack of data from some structures and the described similarities between them were the stimulus for the development of PBSHM.

The PBSHM discipline is developed by trying to deal with a wide range of knowledge-transfer problems, from more simple to more complicated ones. On the simpler side of the problem spectrum, a knowledge transfer problem and an attempt to solve it is

* Corresponding author.
E-mail address: g.tsialiamanis@sheffield.ac.uk (G. Tsialiamanis).

https://doi.org/10.1016/j.ymssp.2024.111119
Received 14 September 2023; Received in revised form 1 December 2023; Accepted 5 January 2024
Available online 19 January 2024

described in [1]. In the aforementioned work, the problem is that of transferring knowledge within a *homogeneous* population. Such a population may result from the procedure of constructing nominally-identical structures or structural members. In this case, the variations in the characteristics of the structures arise from inherently random events during the construction or because of the exposure of the structures to the environment, which may cause small defects. As a result, the structures shall have similar but not identical behaviour.

The approach described in [1] for this type of populations is the definition of a single model to explain the behaviour of each structure within the population. The proposed term for such a model is a *form*, motivated by Plato's Protagoras and Meno [10] and The Republic [11]. For Plato, a form is a set of rules and behaviours that an entity follows, according to which group of beings it belongs to. Similarly, given a population of structures and considering it a group of beings, the behaviour of all the structures within it should follow the corresponding form. The form is not very strict and allows some variations. Likewise all structures do not need to behave exactly the same way, but have some room for divergence between them. To allow such differentiation, an appropriate type of model for a form is a *probabilistic* or a *generative* model. A straightforward example of such a model, which is also used in [1], is a *Gaussian process* (GP) model [12]. The GP provides a conditional *probability density function* (PDF) of a monitored quantity of the structure, given some input variables. As a result, such a form can be trained on an existing and monitored population, and can be used to perform novelty detection for new members of the population, which have not been extensively monitored.

Attempting to move towards a more complicated knowledge-transfer problem, one might find out that one needs to transfer knowledge between populations of structures which are not nominally identical. Populations of such dissimilar structures are referred to as *heterogeneous* populations in [1–5]. The heterogeneity of the populations may vary from quite high levels to very low levels of quite similar, but not nominally-identical, structures. In a quite general case, every structure could be considered part of a single population. In that case, smaller communities of the population can be separated from the whole population, to facilitate the knowledge transfer between them. To assess the similarity of the structures and classify them in subgroups, a similarity metric is required. For this purpose, in [2] a transformation of the structures into *irreducible elements* (IEs) and the formation of graphs using these elements is discussed. In [5], the procedure of transforming structures into IEs and the creation of the corresponding *attributed graphs* (AGs) is extended and a matching algorithm is presented.

The transformation of structures into graphs provides a convenient way of encoding the layout of the structures in a form which is recognisable by *machine learning* [13–15] algorithms. Using standard algorithms or learning algorithms, the graphs can be processed and the similarity of structures can be evaluated. In [2], communities are sought within the population, as well as similar subgraphs within the population. In [5], a machine-learning algorithm is used to classify the structures in communities, which seems to be able to classify specific types of structures following similar logic to human intuition; for example, bridges are classified according to the number of spans they have. In [4], the created graphs are also used as inputs to *graph neural networks* (GNNs) [16] to perform inference within a population of structures regarding a quantity of interest. If the specific quantity, which is predicted from the algorithm, is a damage-sensitive quantity, the model can be used to infer this quantity for newly-built structures, for which acquired data do not exist.

A common problem of SHM, for which data availability is a major issue, is that of damage prognosis [17,18]. Dealing with most prognosis problems is fundamentally an extrapolation problem; although prognosis problems with periodic behaviour exist, it is not the case in general. In the case of damage prognosis, the notion of extrapolation is even more evident, because of the irreversibility of the damage evolution procedure. Of course, structures can be repaired and be once again operational, but it is unlikely that the repaired structure will exhibit the same behaviour as before [19]. Even replacing a damaged component, could result in a structure with different behaviour because of the uncertainty of the connectivity of the new member with the rest of the structure.

To reduce the dependence of data, one could resort to physics-based approaches. Such approaches provide an analytical model of the effect of damage on the behaviour of a structure and the way that damage evolves over time [20,21]. However, evolution of damage in structural members is a quite complicated process and, if one takes into account the uncertainty added because of the environmental and operational conditions, its analytical modelling can become quite demanding. As in most modelling problems, statistical and data-driven approaches are a viable option. Statistical models of degradation processes can be used to make predictions of the remaining useful life of systems [22]. As a more modern approach, machine learning is an alternative, and algorithms such as *recurrent neural networks* (RNNs) [23] could be used to perform damage prognosis.

Although a machine-learning approach seems appealing, such approaches might not be a viable option. Such models require sufficient data to be trained, and, if one wants to build an appropriate model of the damage evolution mechanism, the data should be available until the failure of the structure; or up to a point that the structure needs immediate repair. Given that, as discussed previously, a repaired structure is not the same as before the repair, such data are difficult to acquire. Arguably, one of the most effective ways to acquire such data would be using a population-based strategy. As in [1], the evolution of damage in a nominally-identical population can be modelled using a damage-evolution form. A difference between a damage-prognosis form and the normal-condition form presented in [1] is that a damage-prognosis form should be informed by partial observations about the current state of the tested structure. The information is partial and biased, because the structure's lifetime starts from an undamaged state and evolves towards its degraded states, which is expected, because the acquired data from a monitored structure tend to be from its less damaged states and the desired predictions are about states of higher level of damage.

In the current work, a population-based approach to damage prognosis is studied. The desired approach is the definition of algorithms, which do not require the analyser's knowledge as input to the model. Approaches which would require such knowledge are called *physics-based* and, as discussed, the formalisation of the physics of damage evolution can be quite complicated and highly uncertain. Thus, the approach selected in the current paper is a data-driven one. Moreover, a damage-mechanism model requires quite accurate measurements of quantities that describe the evolution of damage; for example in [21], the XFEM method is used

to model the effect of a crack in the mechanical properties of a body, but to do so, the characteristics of the crack are needed. Measuring such quantities is not always feasible and a method applied directly on the evolution of monitored quantities would be more convenient.

The population-based approach proposed here is based on the belief that a population of nominally identical or quite similar structures will have a similar behaviour regarding damage evolution. It is also considered that, because of the uncertainty of the loading conditions and the exact damage, the behaviour will vary. Therefore, the proposed approaches shall be informed both by data acquired from the structure, while damage evolves, and by the population. The first approach considered is that of applying a *functional principal component analysis* (fPCA) [24] on the population data. By doing so, a subspace of the total space of functions is defined, where one can consider that the functions describing the damage evolution of the population members exist. Having defined that space, and using partial observations of a new structure where damage has started evolving, one can perform a Bayesian search for the function which belongs to the functional subspace of the population and best fits the observed data. As more data are acquired, i.e. as the damage evolves further, more points are included in the inference procedure and the more accurate the provided predictions become.

The second approach is that of using *meta-learning* [25], a recently-developed subdiscipline of machine learning. Although a quite popular interpretation of what meta-learning includes is algorithms that "learn to learn" [25], for population-based approaches of modelling systems a more fitting interpretation of such algorithms is that they draw information from the tasks of the population, in order to perform more accurate predictions for tasks, whose data availability is restricted. Such an approach is suitable for a population-based framework, especially for damage prognosis, where the data for the task at hand are biased and restricted.

The two methods are presented as two alternatives to the population-based approach to structural health monitoring and more specifically to the problem of damage prognosis. The first method is considered as a more direct and explainable approach to defining an underlying functional space of the physics of the population. The explainability comes from the fact that, given such a functional space, one could sample from it and explore the potential behaviour of the structures. The second approach comes from the machine-learning field of meta-learning and is a completely physics-blind approach. According to the second approach, a neural network is assigned with the task of learning the correlation between values of the quantity of interest for different input values within the population. The exploration and explanation of the resulting model might be more challenging than in the first case, however such methods might be more flexible because of learning local correlations instead of a total underlying function space. Both methods are included in the current work as a comprehensive study for PBSHM damage prognosis.

The layout of the paper is as follows. Section 2 describes the effect of applying fPCA on functional data from a population of structures. Section 3 provides an introduction to meta-learning and more specifically to the algorithm which is used here, the *deep kernel transfer* (DKT) [26]. Section 4 describes the application of fPCA for damage prognosis with re results of the application and Section 5 presents the results of applying the DKT method to an experimental crack-growth dataset. Finally, in Section 6 the results and the behaviour of the algorithms are discussed and in Section 7, conclusions are drawn about the methods.

2. Structural data in the functional principal component space

A major problem when modelling a system is uncertainty. Uncertainty is often categorised as of two types, the *aleatory* uncertainty and the *epistemic* uncertainty. The first type refers to inherently random processes and the second refers to the lack of knowledge about the actual way that a system works. Although the above separation of sources and effects of uncertainty is quite convenient and used throughout the literature, in the current work a slightly different naming and separation is considered. The preferred separation herein is that of *uncertainty in the parameters* and *structural uncertainty*.

The uncertainty in the parameters is quite similar to aleatory uncertainty. Common sources of such a type of uncertainty for modelling of structures are the manufacturing process of structural members, environmental conditions and operational conditions. It could be quite complicated to attempt to model or extensively control these procedure and some of the events which form them may be considered totally random. As a result, the actual structural parameters, for example Young's modulus, differ from one's belief or from the specifications of the manufacturing procedure. Such variations could also occur because of environmental conditions and ageing of materials. The main difference in the current definition of uncertainty in the parameters is that several parameters of the formalisation of the model may fall in this category, while aleatory uncertainty is mainly about fundamentally-random quantities. For example, one could be modelling a nonlinear structure using a Duffing oscillator model, i.e. a cubic nonlinearity, however, the behaviour may be better explained using a power which slightly diverges from the third power. Such a type of uncertainty would be considered epistemic according to the previous formalisation, but in the current one it would be in the parameters.

The second type, the structural uncertainty, is fundamentally about the mathematical formalisation that the analysers use to model a system. It has to do with parts of the system or other factors that affect the behaviour of the system and have been omitted or that the way, according to which they affect the system, has not been properly modelled. The difference from epistemic uncertainty is that the structural uncertainty may only contain the part of the uncertainty that cannot be reduced simply by altering the parameters of a model.

The aim of the current work is to model the evolution of damage in a population-based framework relying exclusively on data. This approach is desired to avoid any structural uncertainty from defining a physics-based model for the specific phenomenon. At the same time, the desired approach should be able to deal with uncertainty in the parameters, because of potential noise in the measurements or, as mentioned already, inherently random events during the damage evolution procedure.

In contrast to the approach of the current work, the physics-based approach to modelling a system is to use one's intuition and define a model f^p, which for some input variables $\boldsymbol{x}$ would provide predictions $\boldsymbol{y}$ about the behaviour of the system. Of course, for

the model to be applicable to many different systems, with common physics, the model should have some adjustable parameters $\boldsymbol{\alpha}$. Therefore, the prediction process is given by,

$$\boldsymbol{y} = f^{p}(\boldsymbol{x};\boldsymbol{\alpha}) \tag{1}$$

where $\boldsymbol{y}$, $\boldsymbol{x}$ and $\boldsymbol{\alpha}$ are vectors of variables and parameters.

Because of the use of tunable parameters $\boldsymbol{\alpha}$ as part of the function, the potential functions, that can be selected to explain the behaviour of a system, define a family of functions $\mathcal{F}^{p}$. This family defines all the potentially-observable behaviours of every system in the studied family of systems. Obviously, this applies to a population of structures. Phenomena, which are modelled within a population-based framework, may follow Eq. (1). Consequently, one has to define a proper family of functions $\mathcal{F}^{p}$, which is a functional of all the potential behaviours of the members of the population.

Given that the functional family $\mathcal{F}^{p}$ is defined following one's intuition and understanding, the modelling framework is susceptible to structural uncertainty. Trying to mathematically formulate what this uncertainty means, a second functional family has to be defined, i.e. the functional family of the *actual* behaviours of the systems $h \in \mathcal{H}$. The general case will be that $\mathcal{F}^{p} \not\equiv \mathcal{H}$, as a result, the predictions of the models f^{p} will diverge from the actual behaviour of the structures of interest.

In the case of $\mathcal{F}^{p} \subset \mathcal{H}$, then there is no structural uncertainty. Such a case may not be very common in a physics-based approach, as it would require a quite general family of functions $\mathcal{F}^{p}$, but for a data-driven approach, a Gaussian process or a neural network are models that can potentially achieve such a condition; since a very general prior is considered and with every observation some functions are rejected, resulting in a smaller set of functions with every observation. In the case of $\mathcal{F}^{p} \subset \mathcal{H}$, the only uncertainty that exists is that of the parameters, and what is needed for accurate predictions is informative (and potentially noise-free) data to select the most fitting parameters for the case.

Data from the system or structure whose modelling is desired are not always accessible. As a result, the definition of $\mathcal{H}$ is not always feasible. A major example is modelling of a structure, which has not been deployed yet. In that case, the use of existing knowledge is the only way to perform modelling. However, in this case, the knowledge may still be considered population-motivated, since it comes from studying past similar structures or structural members. Motivated by this observation, the current work aims at a data-driven definition of population-based models. All things considered, it might be argued that most types of modelling are data-driven, considering that even calibrating a physics-based model fits the *empirical risk minimisation* (ERM) procedure, which is described in [27] as a core element of data-driven learning, i.e. finding the most fitting parameters α for a model of the form of Eq. (1) regardless of the way that the model has been defined, physics-based or data-driven.

The motivation of the current work is that for a population, a proper family of functions $\mathcal{H}$ can be defined via the available data from existing structures. To do so, *functional principal component analysis* (fPCA) is used [24]. To follow such an approach, the available data should be in the form,

$$D = \{h_1, h_2..., h_N\}, \quad h_i = \{(\boldsymbol{x}_1, y_1^i), (\boldsymbol{x}_2, y_2^i)..., (\boldsymbol{x}_M, y_M^i)\}, \quad i = 1, 2..., N \tag{2}$$

where D is the available dataset, h_i is a set of observations for structure i of the form $\{(\boldsymbol{x}_1, y_1^i), (\boldsymbol{x}_2, y_2^i)..., (\boldsymbol{x}_M, y_M^i)\}$, where x_j is the jth input vector and y_j^i is the jth output variable of the ith structure. Using this dataset and fPCA, a linear subspace $\mathcal{F}^{fPCA}$ of the space of all possible functions is defined. Given that the data are representative of the population (the evaluation of which is not the objective of the current work), the potential behaviour of all structures belonging to the same populations and under the same damage, environmental and operational conditions live in this subspace. In this case, it can be considered that $\mathcal{H} \subset \mathcal{F}^{fPCA}$. The functions $f_i \in \mathcal{F}^{fPCA}$ are given by,

$$f_i(\boldsymbol{x}) = \sum_{j=1}^{j=K} \beta_j^i \varphi_j(\boldsymbol{x}) + \epsilon, \ \boldsymbol{x} \in \mathcal{X}, \ f_i \in \mathcal{F}^{fPCA} \tag{3}$$

where $\varphi_j(\boldsymbol{x})$ are the basis functions of the fPCA decomposition of the data, β_j^i are the coefficients of the basis functions used to compose f_i, and K is the number of principal components considered in the decomposition in order to maximise the explained variance. To obtain the principal component basis-functions $\varphi_j(\boldsymbol{x})$, using a set of functional observations h_i, a PCA approach can be followed for discrete data or an fPCA approach as described in [24]. Given some partial observations of the behaviour of a new structure in the population, the problem of predicting the behaviour for the whole input space is reduced to selecting the most fitting principal component scores $\boldsymbol{\beta_j}$. The search for the most fitting parameters can be performed with several algorithms. The efficiency depends largely on the dimensionality K of the problem. In the current work, a Bayesian Hamiltonian Monte Carlo [28,29] approach is considered for this search.

The belief that the actual subspace of functions $\mathcal{H}$ is a subset of the fPCA space $\mathcal{F}^{fPCA}$ comes from the fact that fPCA is a linear decomposition designed to contain as much information as possible from the original functions. Moreover, it is an infinite space, because the values of the principal component scores β_j may be any real number. In fact, the actual subspace $\mathcal{H}$ might be a manifold in the total functional space and in the fPCA space as well. Such an issue will be further discussed in Section 4, where a specific problem of damage prognosis is presented and the fact that the data actually exist in a manifold of $\mathcal{F}^{fPCA}$ becomes evident.

3. Meta-learning

A recently emerging class of algorithms in the machine-learning community has been that of meta-learning. Several methods are considered to belong in the specific field, a comprehensive review of which can be found in [25,30]. A common characteristic of the

methods is that they are essentially population-based approaches to the development of machine learning models. Looking further into the functionality of the algorithms, the field of meta-learning could be characterised as a special case of *transfer learning* [31] and, in many cases, of *multi-task learning* [32].

The class of meta-learning algorithms which are of interest herein are the ones that draw information from a population of tasks, which, for the interests of structural dynamics and structural health monitoring, are often separate structures. It should be noted that the desired type of model is a common model for the whole population, which can be with a few samples adjusted to a member of the population, given some data from it. Such a framework is motivated by the functionality of physics-based models, such as finite element models (FEM) [33], which have a few tunable parameters and, by varying them, the behaviour of many different members of the population can be modelled.

As described in [30], there are several types of meta-learning algorithms. One of these types is focussed on the parameter initialisation of the model. The idea is quite appealing for neural network approaches, whose number of tunable parameters is quite high. An approach of defining such a model is described in [34], where the search for an initialisation point is performed, setting as a criterion that, from the initialisation point, the training algorithm (backpropagation in this case) shall be able to converge to a satisfactory point with a few training steps and a few samples. The algorithm is built as a two-step backpropagation algorithm, whose first step takes into account the errors of individual tasks from the population and the second step attempts to exploit information from the first step to find the desired initialisation point. The method is compared to a simple average-point initialisation approach, i.e. simply averaging the minima points of all tasks of the population, and seems to be more efficient.

Another approach to meta-learning is that of targeting the learning algorithm to make it case-specific and more efficient for a population of tasks. An implementation of such an approach is presented in [35]. In the aforementioned work, the learning algorithm is replaced by a neural network. The idea behind such an approach is that the learning algorithm and the loss functions used in optimisation of neural networks are often quite general and not task-specific. Therefore, training a neural network as a population-specific neural-network trainer seems an appealing idea for a population-specialised optimiser. The results also reveal that the method provides faster convergence and with lower error.

An interesting approach is presented in [36]. The method, called *conditional neural processes* (CNP), aims at defining variables that characterise the tasks. The idea is quite appealing in the case of physical problems, since, in a heterogeneous population, there might be some underlying variables that characterise the space of structures. These variables may be difficult to define, therefore an approximation of these via the use of a secondary neural network is attempted in the CNP framework. The secondary neural network uses available data (both inputs and targets), as inputs and predicts a latent vector, which is then used as an input to the main predictive neural network. The framework aims at training the secondary neural network to provide task-informative vectors, so that the main predictive neural network shall be able to make more accurate predictions about new data-poor tasks of the population.

Other approaches include sharing of features and some *few-shot learning* approaches [37]. In the current work, the preferred method is that of deep kernel transfer (DKT) [26]. The method is selected because of its simplicity, the reasonable intuition available on the way the method works, and because of the use of Gaussian processes, which is a Bayesian algorithm and is able to properly treat uncertainty.

Deep kernel transfer is built on top of the Gaussian process training framework. More specifically, a Gaussian process model considers as inputs data D of the form,

$$D = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2)..., (\boldsymbol{x}_M, y_M)\} \tag{4}$$

where $\boldsymbol{x}_i$ is the ith sample input vector, y_i is the ith sample target output, and M is the number of observations. Consequently, a functional prior is selected for the candidate functions, which shall be considered for the interpolation of the data in D. The prior is given by,

$$f(x) \sim \mathcal{GP}(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')) \tag{5}$$

where $m(x)$ is the mean function of the prior, which is a function of the input vectors $\boldsymbol{x}$ and $k(\boldsymbol{x}, \boldsymbol{x}')$ is the kernel function, which is calculated between two input vectors $\boldsymbol{x}$ and $\boldsymbol{x}'$.

Taking into consideration the data D, one can make predictions about new input points $\boldsymbol{x}^*$. The predictions are given by,

$$f(\boldsymbol{x}^*|D) \sim \mathcal{N}(\bar{f}^*, cov(f^*)) \tag{6}$$

where $\mathcal{N}$ is a Gaussian distribution, and $\bar{f}^*$ and $cov(f^*)$ are the mean and covariance of the predictions for $\boldsymbol{x}^*$, which can be, in some cases, calculated analytically as shown in [12]. The hyperparameters often included in the process are a scale factor, a length scale of the kernel, and the variance of the noise, in case it is considered that the data are noisy. The selection of the hyperparameters can be performed via *likelihood maximisation* or *variational inference*.

The selection of the mean and kernel functions is performed by the analyst. Common options are a zero-mean function or a linear one, and an exponential or Matern kernel. These selections reflect the belief that one has that the about the underlying physics of the system which is modelled. The mean function may reflect one's belief that a trend exists in the data and the kernel may reflect the smoothness or the periodicity of the data. Proper selection of such parameters can be performed to induce physical knowledge in the algorithm; such examples are presented in [38].

The main addition to the traditional GP definition by the DKT framework is that in the second case the data are available in a population-based framework,

$$D^i = \{(\boldsymbol{x}_1^i, y_1^i), (\boldsymbol{x}_2^i, y_2^i)..., (\boldsymbol{x}_M^i, y_M^i)\}, \quad i = 1, 2..., N \tag{7}$$

where $\mathcal{D}^i$ is the dataset of the ith task/structure, M is the number of observations and N is the number of structures. Another difference is that, in the second case, the mean and the covariance functions are learnt instead of being imposed by the analyser. To do so, a neural network is used. The neural network $F_{\boldsymbol{\phi}}$ is used to preprocess the input vectors $\boldsymbol{x}$ before they are fed into the mean function and the kernel function. Thus, the new prior is given by,

$$f^i(x) \sim \mathcal{GP}(m(\mathcal{F}_{\boldsymbol{\phi}}(x)), k(\mathcal{F}_{\boldsymbol{\phi}}(x), \mathcal{F}_{\boldsymbol{\phi}}(x'))) \tag{8}$$

where $\boldsymbol{\phi}$ are the trainable parameters of the neural network. The mean and kernel functions can be used for the Gaussian process and Eq. (6) gives the predictions for new points $\boldsymbol{x}^*$ following,

$$f^i(\boldsymbol{x}^*|D) \sim \mathcal{N}(\bar{f}^*_{\boldsymbol{\phi}}, cov(f^*_{\boldsymbol{\phi}})) \tag{9}$$

but in this case the mean and covariance of the normal distribution of the predictions are functions of the trainable parameters of the neural network $F_{\boldsymbol{\phi}}$. Moreover, a loss function is considered, which is common for every task. The loss function is given by,

$$\mathcal{L}^i = -\log p_{\phi,\varphi}(y^i|\mathcal{D}^i) \tag{10}$$

where $p_{\boldsymbol{\phi},\boldsymbol{\varphi}}$ is the probability density function predicted by the model, which is a function of the tunable parameters of the neural network ϕ and of the parameters $\boldsymbol{\varphi}$ of the GP. The loss function is aimed at maximising the probability of the target values y^i of the data $\mathcal{D}^i$. During every training loop, for every task i, the error $\mathcal{L}^i$ is calculated and then this error is backpropagated to perform an optimisation training step for parameters $\boldsymbol{\phi}$ of the neural network F and the parameters $\boldsymbol{\varphi}$ of the GP. More specifically, the backpropagation step is given by,

$$\boldsymbol{\phi}' \leftarrow \boldsymbol{\phi} - \gamma_1 \nabla_{\phi} \mathcal{L}^i \tag{11}$$

where $\boldsymbol{\phi}'$ are the updated values of the neural-network parameters $\boldsymbol{\phi}$ after one training iteration, γ_1 is the learning rate of the neural-network parameters and,

$$\boldsymbol{\varphi}' \leftarrow \boldsymbol{\varphi} - \gamma_2 \nabla_{\varphi} \mathcal{L}^i \tag{12}$$

where $\boldsymbol{\varphi}'$ are the updated values of the Gaussian-process parameters $\boldsymbol{\phi}$ after one training iteration, γ_2 is the learning rate of the Gaussian-process parameters. Although Gaussian processes are often considered a non-parametric algorithm, their hyperparameters need tuning. These are often the correlation length of the kernel, the noise parameter of the process and the scale of the kernel. In some cases more parameters can be included in this optimisation procedure, e.g. for a linear mean function, the slope of the function. After completing training, the resulting parameters $\hat{\boldsymbol{\phi}}$ and $\hat{\boldsymbol{\varphi}}$ from the optimisation algorithm are identified. For a new structure, observations D^* similar to Eq. (7) are acquired and, following the traditional GP framework, the predictions for the labels y^* for $\boldsymbol{x}^*$ are given by,

$$p(y^*) \sim GP(m(\mathcal{F}_{\hat{\boldsymbol{\phi}}}(\boldsymbol{x}^*)), k_{\hat{\boldsymbol{\varphi}}}(\mathcal{F}_{\hat{\boldsymbol{\phi}}}(\boldsymbol{x}^*), \mathcal{F}_{\hat{\boldsymbol{\phi}}}(\boldsymbol{x}'^*))|D^*) \tag{13}$$

Considering every structure to be a different task, the current application follows the methodology defined in [26], where the code for the algorithm is presented.

The intuition behind such an algorithm is that it shall learn a population-informed transformation of the input variables, so that it can perform effective inference for all the tasks. The transformation that the network $F_{\hat{\boldsymbol{\phi}}}$ provides may have several forms. In a quite high-dimensional problem, the network may perform a feature-extraction procedure in order to identify informative features so that the GP can afterwards perform inference. In more simple and low-dimensional problems, like the one which is presented here, the neural network is expected to transform the input space, so that the mean and kernel functions of the GP are effectively informed by the population. This transformation might be a deformation of the input variables $\boldsymbol{x}$, so that a linear mean function and an exponential kernel represent more accurately the actual mean function of the population and the correlation between points of the input space and the target quantities.

Following a data-driven path instead of defining the two quantities manually, is expected to deal with the structural uncertainty issue. When modelling structures, it is expected that analytical solutions for specific problems may not be available. The problem might be highly evident in high-dimensional problems, where the definition of a formula may be very difficult, even in the form of a prior belief. Therefore, it is desired that machine-learning algorithms are used to extract such relationships, at least at an approximation level. For damage prognosis, as discussed, the damage-evolution phenomenon is difficult to model and involves high levels of uncertainty. As a result, the population-based DKT approach is considered a potential solution to the problem.

4. Application of fPCA to crack-propagation

4.1. Experimental set-up

To test the efficiency of estimating the actual functional space of a population phenomenon, in the current work, a population of plates is chosen and the phenomenon is the growth of a crack in them. The population comprises six aluminium plates, which were subjected to cyclic loading in a laboratory environment; one of the plates is shown in Fig. 1. The damage is a skin crack and was artificially initiated in the centre of the plates; the value of the initial crack was 16 mm. The load was a sinusoidal load of 12 Hz frequency with maximum amplitude of 35kN and a load ratio of $R = 0.1$. The crack is manually measured with calipers during the

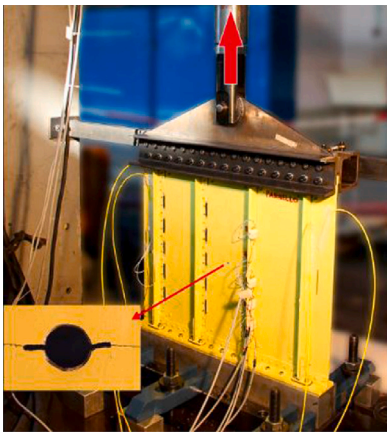

Fig. 1. The experimental setup used to acquire the data of the current application [39].

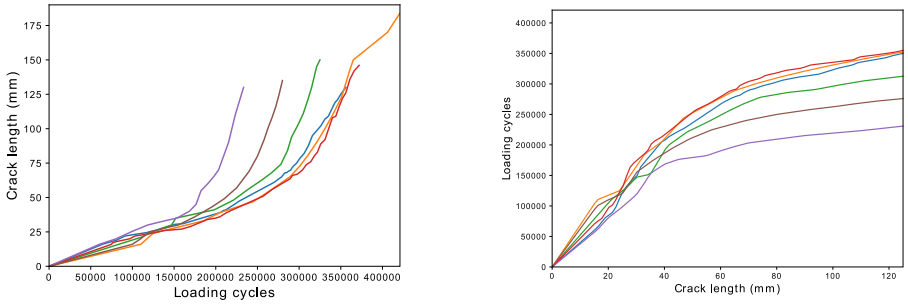

Fig. 2. The crack-growth paths of the six plates in their original form (left) and with inverted axes (right), for the purposes of the current work. For reference, the colours of the plates corresponding to numbers one to six is considered to be blue, orange, green, red, purple, brown. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

system operation. The plates are representative of plates of the fuselage of a helicopter. For more information, the interested reader can refer to [39,40].

The crack-growth curves of the six plates are shown in Fig. 2 on the left. It immediately becomes evident that, although the plates are considered nominally identical and form a homogeneous population, their behaviour, regarding the way that the crack grows, is quite different. The crack requires different numbers of loading cycles to reach similar values of crack lengths for every plate. This difference in the behaviour is quite evident, especially after the cracks achieve a length of around 30-40 mm.

4.2. Definition of the fPCA subspace

In order to apply the methodology described previously, the available dataset should have a functional form; i.e. the available curves should be functions and, thus, there should exist a common range of each function. This clearly does not happen when one considers the dataset in the form of Fig. 2 on the left. However, considering the modelled relationship a function which maps the crack length to the required number of loading cycles (instead of the opposite), a functional relationship is defined. By defining an upper limit for the crack length, and inverting the axes for the aforementioned reason, the result is shown in Fig. 2 on the right.

The latter form of the data is particularly useful, because, in this form, fPCA and data augmentation are applicable. The result of applying fPCA is the definition of the principal components of the functional relationship between the crack length and the loading cycles. These components are shown in Fig. 3 on the left and the principal component scores of each curve of Fig. 2 are shown on the right. Using two principal components, 99% of the variance of the data is explained. What is interesting in the aforementioned figures is that the behaviour of the cracks contains two components, in which one is almost steadily decreasing but the second has a peculiar behaviour. Therefore, one could speculate that the accurate physical modelling and interpretation of the specific data would be quite difficult, since physical intuition would probably omit inclusion of such behaviour in an analytical model. A second observation is that the principal component scores which correspond to the available data do not spread along the whole $\mathcal{H}^{fPCA}$ space, which would be the whole plane of the right-hand plot. In contrast, they are concentrated in a very small area. This observation indicates that fPCA may identify a space which contains the actual space of functions of the modelled phenomenon.

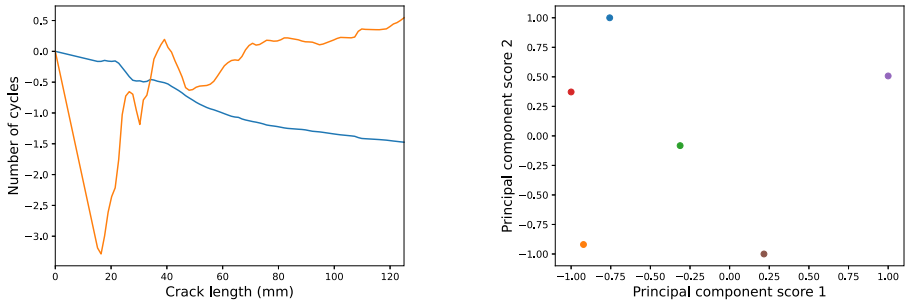

Fig. 3. The principal component functions of the fPCA - φ_j in Eq. (3) - applied on the functional crack-growth data (left). The principal component scores - β_j^1, β_j^2 in Eq. (3) - of the six available curves, with colour correspondence with Fig. 2 (right). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

As mentioned, the specific functional form of the data provides the analyst with the option to augment the dataset. The augmentation is simply performed by defining new curves using the convex combination,

$$c = \alpha c_1 + (1-\alpha)c_2 + \epsilon \tag{14}$$

where c_1 and c_2 are two of the available curves, i.e. they are two distinct samples of h_i of Eq. (2), α is a random variable sampled uniformly from the interval $[0, 1]$, and ϵ is a noise parameter sampled from a positive-only distribution (because the crack may only become larger), a beta distribution in the current work. If specific knowledge about the phenomenon of the crack growth is available, the user could impose a specific distribution for α, but in the current work, a physics-agnostic approach is desired, so the uniform distribution is chosen. This approach in the principal component space, since it is the result of a linear projection of the original curves, is interpreted as the definition of line segments that connect the points of the available data and adding noise to them.

Because of the small number of structures in the population in the current case, the augmentation is required to define a manifold in the functional space. This manifold is considered to be the space of the potential behaviour of the members of the population. The approach is an attempt to define this space in a data-driven manner, rather than trying to identify it using knowledge and analytical solutions. After the space has been defined, for a new structure, where damage starts evolving, an optimisation procedure can be followed to identify the principal component scores that best fit the so-far observations of the damage evolution.

4.3. HMC for damage prognosis

A Bayesian framework can be followed to identify the principal component scores that best fit the data. More specifically, for a structure i of the population, a set of observations $\mathcal{D}^i$ is considered in the form,

$$\mathcal{D}^i = \{(x_1, y_1), (x_2, y_2)..., (x_M, y_M)\} \tag{15}$$

where for the current problem x_j is the jth observed value of crack length, y_j is the number of loading cycles required to achieve the specific crack length and M is the number of observations.

The variables that need to be identified for the specific problem are the principal component scores β_j of Eq. (3). In the current case, because two principal components explain the dataset, the variables that are identified are β_1 and β_2. The likelihood of the two variables is chosen to be Gaussian, the two variables are considered to be independent, and, taking into account the observations of Eq. (15), the likelihood is given by,

$$L(\mathcal{D}^i|\beta_1, \beta_2) = \prod_{j=1}^{M} \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{1}{2}[\frac{(\beta_1\varphi_1(x_j) + \beta_2\varphi_2(x_j) - y_j)}{\sigma}]^2 \tag{16}$$

where σ is the standard deviation of the likelihood and can be included in the learning process or be defined by the analyser, and φ_1 and φ_2 are the principal components, examples of which are shown in Fig. 3 on the left.

Using Bayes theorem, the posterior probability or the log-posterior probability of the parameters can be calculated by,

$$P(\beta_1, \beta_2|\mathcal{D}^i) = \frac{L(\mathcal{D}^i|\beta_1, \beta_2)p(\beta_1, \beta_2)}{p(\mathcal{D}^i)} \sim \log L(\mathcal{D}^i|\beta_1, \beta_2)p(\beta_1, \beta_2) \tag{17}$$

where L is the likelihood given by Eq. (16), $p(\beta_1, \beta_2)$ is the prior belief about the values of β_1 and $\beta 2$ and $p(\mathcal{D}^i)$ is the probability of the data.

The prior belief has been acquired from the population data. More specifically, the prior belief about the values of the parameters β_1, β_2 is the result of the augmented data, which are the result of Eq. (14). To define a probability density function of these parameters, a *normalising flow* (NF) [41] is used. This option may not be the most computationally-efficient for a two-dimensional case, as the one of this example, but NFs are powerful models which can learn quite complicated distributions and would be efficient

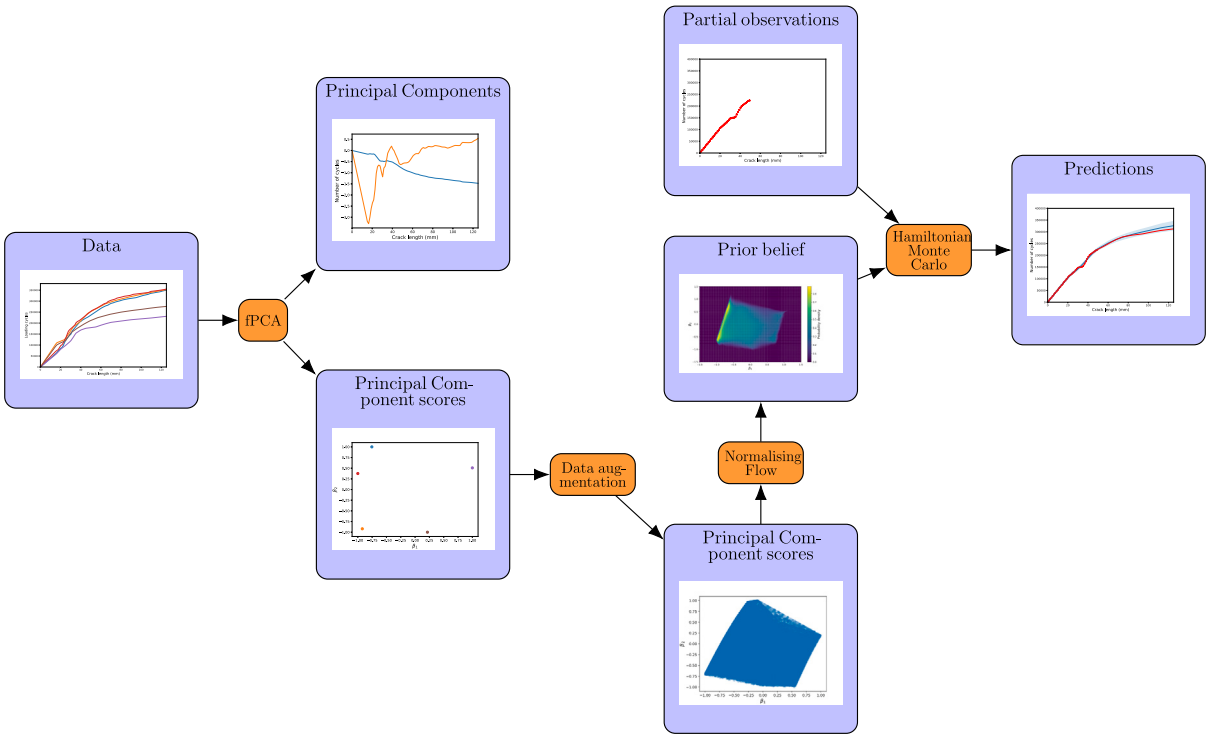

Fig. 4. Schematic procedure of the proposed fPCA algorithm for population-informed inference.

in cases of higher number of principal components. Further explanation of the method of normalising flows is not given here, but the interested reader could refer to [42,43].

For completeness, the models used were *masked affine flows* [44] and the characteristics of the models used in every case were a Gaussian latent distribution with a unit covariance matrix, and an NF with eight flows (i.e. eight neural network transformations from the latent distribution to the target distribution), each of these flows was a neural network with two-dimensional inputs and outputs (because the data are two-dimensional) and a 32-node hidden layer. The models were trained for 20000 epochs with a batch size of 512 samples. Although it is a machine learning model and a validation procedure should be followed to select the hyperparameters of the model and the best model throughout the training procedure, the aim of this NF is auxiliary and its purpose is to model the probability density of the population samples. As a result, the model is selected solely using the normalising-flow training loss and picking the one with the lowest such value.

In order to perform inference, the logarithm of the posterior is used and a *Hamiltonian Monte Carlo* (HMC) algorithm [28], more specifically a No-U-turn algorithm [45]. The implementation of the algorithm which is used here can be found in [46]. The application of such an algorithm for the values of the latent space are also proposed in [47]. Such an algorithm is applicable in the current case, because the posterior of Eq. (17), from which sampling is required, is differentiable. The benefit of using an HMC approach instead of more classic Metropolis–Hastings Markov chain Monte Carlo (MCMC) is that it is more efficient in high-dimensional search spaces. The current problem is quite low-dimensional, however, similarly to using an NF, the algorithm is built for a higher number of principal components. The characteristics of the HMC algorithm in the presented applications are a step size equal to 0.002, 30 samples per step of the algorithm, a burn-in period of 250 samples, and a total of 1250 samples. After using HMC to sample from the posterior, Eq. (3) is used to reconstruct the potential damage paths and to calculate the time until the crack reaches the user-set limit. The procedure is shown schematically in Fig. 4 and a pseudoalgorithm is presented in Algorithm 1.

Observing the fPCA component scores shown in Fig. 3 and the augmented data in Fig. 4, it becomes clear that the data-augmentation process followed here creates a manifold in the PC space by connecting together couples of points in the dataset. Such an augmentation would not be necessary for a larger dataset, with much more dense data points in the PC space. However, in the current case, given the small size of the dataset, the augmentation is necessary. A problem that might arise with a small dataset, like the one available here, is that the specific augmentation process might not create an appropriate manifold in the sense that it might not include proper PCs for a testing damage path. This is evident in the current case, if one observes the right-hand side of Fig. 3. In this plot, it is clear that, if one damage path is considered the testing datum and a manifold is created following the proposed data-augmentation strategy using the other five paths, the created manifold would not include the left-out path. This is true except for the case of considering as testing path the third path (green curve and point).

In the case that the true underlying pair of principal component scores (PCs) does not belong in the manifold of the augmented data, the algorithm is still expected to perform well. Such an expectation comes from the fact that, although the true underlying

Algorithm 1 FPCA subspace inference.

Require: $D = \{h_1, h_2 ..., h_N\}$ (Eq. (2))
Require: Testing structure data $\mathcal{D} = \{(x_1, y_1), (x_2, y_2 ..., (x_m, y_m)\}$
Require: $\mathcal{N}$ number of HMC samples
Require: σ variance of the likelihood
procedure HMC PREDICTIONS($D, \mathcal{D}, \mathcal{N}, \sigma$)
$D_{aug} \leftarrow dataset\_augmentation(D)$ ▷ Eq. (14)
$PCA_{basis}, PCA_{sc} \leftarrow fPCA(D_{aug})$ ▷ PCA_{basis} contains φ_j of Eq. (3)
$NF \leftarrow train\_normalising\_flow(D_{aug})$ ▷ According to [42]
$prior \leftarrow NF$
$likelihood \leftarrow \prod_{j=1}^{m} \frac{1}{\sigma\sqrt{2\pi}} \exp -\frac{1}{2}[\frac{(\beta_1 PCA_{basis}[1](x_j)+\beta_2 PCA_{basis}[2](x_j)-y_j)}{\sigma}]^2$
$[\boldsymbol{\beta}^1, \boldsymbol{\beta}^2 ..., \boldsymbol{\beta}^{\mathcal{N}}] \leftarrow HMC(likelihood, prior)$ ▷ According to [46]
$predictions \leftarrow inverse\_PCA([\boldsymbol{\beta^1}, \boldsymbol{\beta^2} ..., \boldsymbol{\beta}^{\mathcal{N}}]^T)$
return *predictions*
end procedure

pair of β_1 and β_2 does not belong to the manifold, the NF does not assign an absolute zero value of probability density to the rest of the PC space. As a result, the HMC algorithm should potentially be able to locate the appropriate principal component scores. Therefore, for completeness, the algorithm is tested on two problem set-ups. The first is an in-population prediction framework, i.e. all the damage paths are considered in the prior-definition process and data from the augmented dataset are randomly sampled and the performance of the algorithm is tested according to the later samples. The second scenario is an out-of-population prediction framework, where six scenarios are considered. Each scenario is simply defined by using five of the original damage paths as the training population and the sixth as the testing datum.

4.4. In-population predictions

The proposed algorithm is tested on partial data coming from damage paths from the augmented dataset. The augmented data are created using Eq. (14). The testing data in this case are considered to belong to the same population as the training data and the accuracy of the algorithm is expected to be higher.

The testing population comprises 1000 damage-path samples. The results are presented in terms of the *normalised mean-square error* (NMSE) of the final prediction of the predictions of the remaining damage path. The NMSE of the performance of the algorithm is defined by,

$$NMSE = \frac{100}{M\sigma_y^2} \sum_{i=1}^{M} (\hat{y}_i - y_i)^2 \tag{18}$$

where $\hat{y}_i$ is the prediction of the model for the ith damage path, y_i is the corresponding true value of the true remaining damage path, σ_y is the standard deviation of the damage paths throughout the population and M is the total number of samples. Such an error metric is useful because it is equal to 100% if the model predictions ($\hat{y}_i$) are set to the mean value, i.e. $\hat{y}_i = \overline{y}$; values lower than 100% reveal that the model is indeed capturing correlations in the data. Experience with this NMSE indicates that good models are obtained for values of less than 5%, with a value of less than 1% for excellent models.

The results are presented in Figs. 5 and 6 in terms of the mean value and the standard deviation of the NMSE across the testing population. In the specific figures, the horizontal axis represents the crack length, for which the loading cycles are predicted, and the vertical axis represents the current crack length. The colour of the plot in Fig. 5 represents the NMSE of the testing population regarding the prediction of the number of cycles needed to achieve the crack-length value defined by the x axis, given crack-length and number-of-cycles observations up to the crack-length value defined by the y axis. Similarly, in Fig. 6 the colour represents the standard deviation of the NMSE within the testing population. It is clear that as the crack length increases, the mean NMSE of the predictions becomes quite low. It is worth noting that a part of the plots in Figs. 5 and 6 refer to predicting the past, which might seem peculiar. However, this metric is useful as well, because it indicates how well the algorithm is able to explain the so-far data and as a result it could be a metric of whether the testing structure indeed belongs to the population.

4.5. Out-of-population predictions

The results of applying the proposed algorithm to make predictions about one plate, considering the other five plates as training population, are summarised in Figs. 7 and 8. In the aforementioned figures, it is clear that the predictions get more accurate as the crack grows and more observations are included in the inference process. More plots of how the predictions evolve as more data are acquired are provided in the appendix and the results are further discussed in Section 6. In the plots in Fig. 8, it becomes

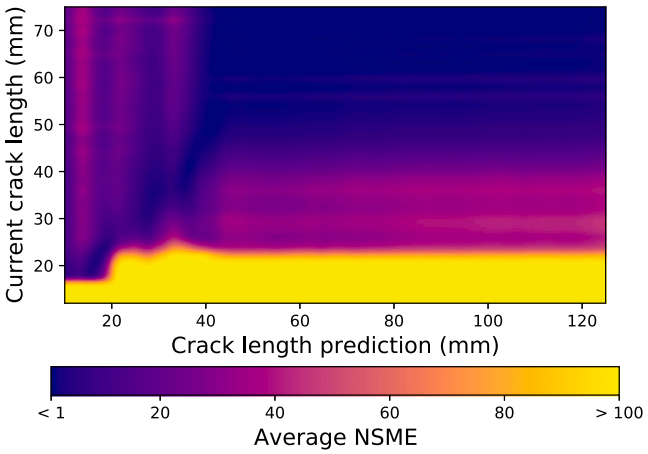

Fig. 5. Average normalised mean-squared error of the predictions of the fPCA model for a testing population of 1000 structures. The horizontal axis represents the crack length to which the predictions refer and the vertical axis represents the current crack length.

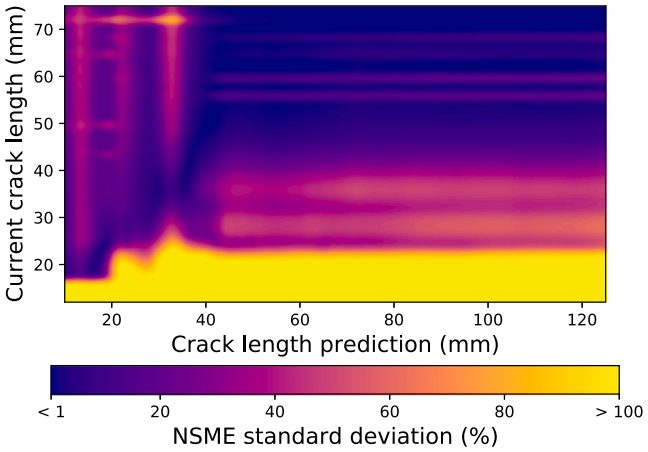

Fig. 6. Standard deviation of the normalised mean-squared error of the predictions of the fPCA model for a testing population of 1000 structures. The horizontal axis represents the crack length to which the predictions refer and the vertical axis represents the current crack length.

clear that in some cases the algorithm is not able to fit quite well in the existing data, with an exception of plates one and six. This observation could indicate to the analysts that the specific plates should be studied using a different approach. Indeed for plate five, the predictions for the past are not satisfactory and so do the predictions for the future for the most of the crack-growth procedure.

5. Application of meta-learning to crack-propagation

The DKT meta-learning approach, which was described in Section 3, is also tested on the aforementioned crack-growth dataset. For the application, similarly to before, a training population is assumed and the model is trained based on data acquired from these structures. Subsequently, the trained model is applied on the testing population or a single testing structure. The predictions of the model are calculated using Eq. (13). The procedure is schematically shown in Fig. 9.

5.1. In-population predictions

Similar to the fPCA approach, the meta-learning algorithm is tested on the two cases of in- and out-of- population predictions. For the case of making predictions for structures that belong to the population, similar to before, 1000 damage paths generated from Eq. (14) are considered. Because the results refer to a testing population, the accuracy of the algorithm is evaluated in terms of the average NMSE and the standard deviation of the NMSE across the testing population.

The average NMSE of the predictions is shown in Fig. 10. Similar to before, for a crack length of above 40 mm, the average NMSE is almost always below 1%. Similar behaviour is observed for the standard deviation of the NMSE of the predictions in Fig. 11, indicating a quite consistent behaviour across the testing population, which is desired from such algorithms. Regarding the so-far predictions, the algorithm has really high accuracy, indicating that it shall have high accuracy for future predictions as well.

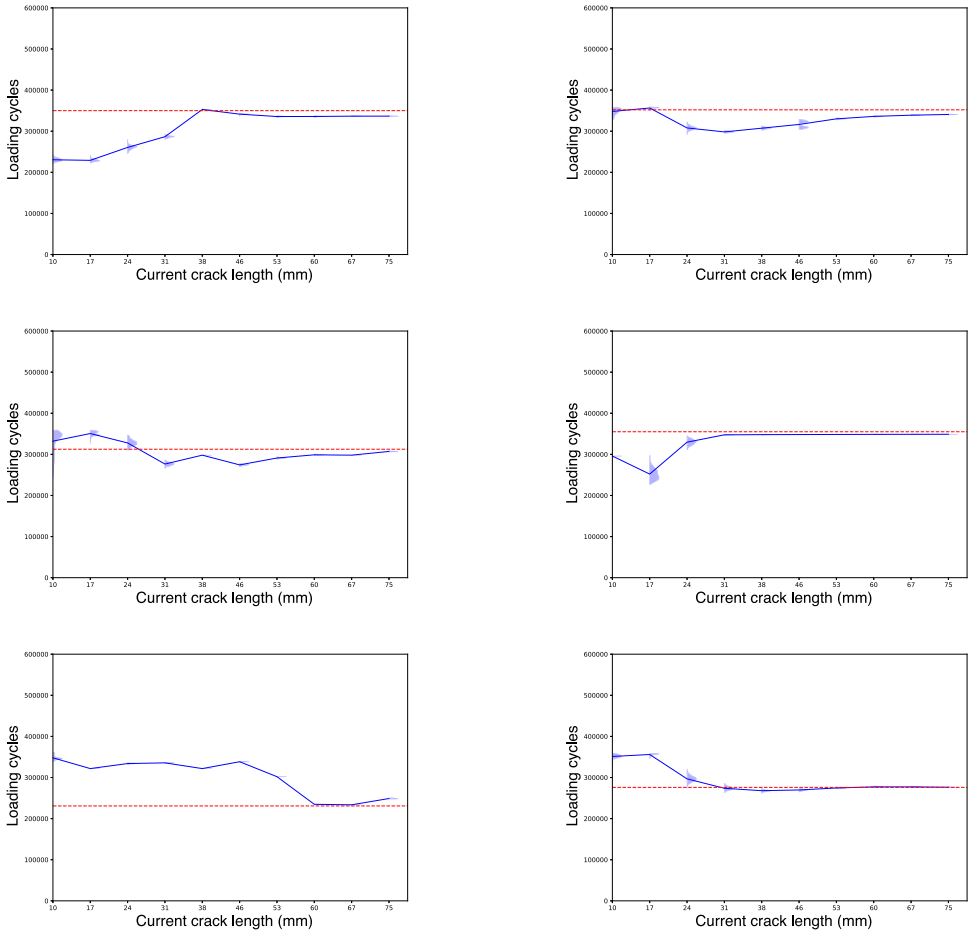

Fig. 7. The PDFs of the estimated total lifetime of the structures using the fPCA algorithm, as a function of the current crack length. The mean values of the predictions are shown as the blue curves and the actual total life-time of each plate is shown as the horizontal red line. The numbers of the plates are one to six from right to left and from top to bottom.

5.2. Out-of-population predictions

To evaluate the efficiency of the algorithm, the same six single-testing-plate problems are considered. The results of applying the DKT algorithm are described by Figs. 12 and 13. In the aforementioned figures, it is clear that as, more information is acquired, the predictions become more accurate and concentrate around the actual remaining useful life of each plate. The algorithm is able to make accurate predictions even for out-of-population plates, implying that the neural network of the DKT algorithm has learnt an appropriate transformation of the input space, so that the mean and the correlation functions resemble the actual mean and correlation of the population. More figures for a more comprehensive qualitative evaluation of the algorithm are provided in the Appendix and a discussion regarding the results is carried out in the next section. In the case of the DKT algorithm, in contrast to the fPCA algorithm, the accuracy for the so-far predictions is satisfactory in most cases. This observation indicates that the algorithm is more efficient in learning the actual underlying physics from the training population and applying this knowledge to future predictions.

6. Discussion

The results show that the two methods presented are able to exploit information from a population of data-rich structures to make predictions about new structures, whose acquired datasets suffer from data scarcity. Observing the results, it becomes clear that both methods become more accurate as more data are acquired. For both types of testing data – in- and out-of- population – the methods are able to make more accurate and more certain predictions as more crack-growth data are observed.

The increasing accuracy of the methods as more data are observed is especially clear for the in-population case studies. The characteristic of these case studies is that the testing population is similar to the training population (i.e. the inference problems come from the same distribution of tasks). Although this is a facilitating assumption, it might be quite common that such inference is

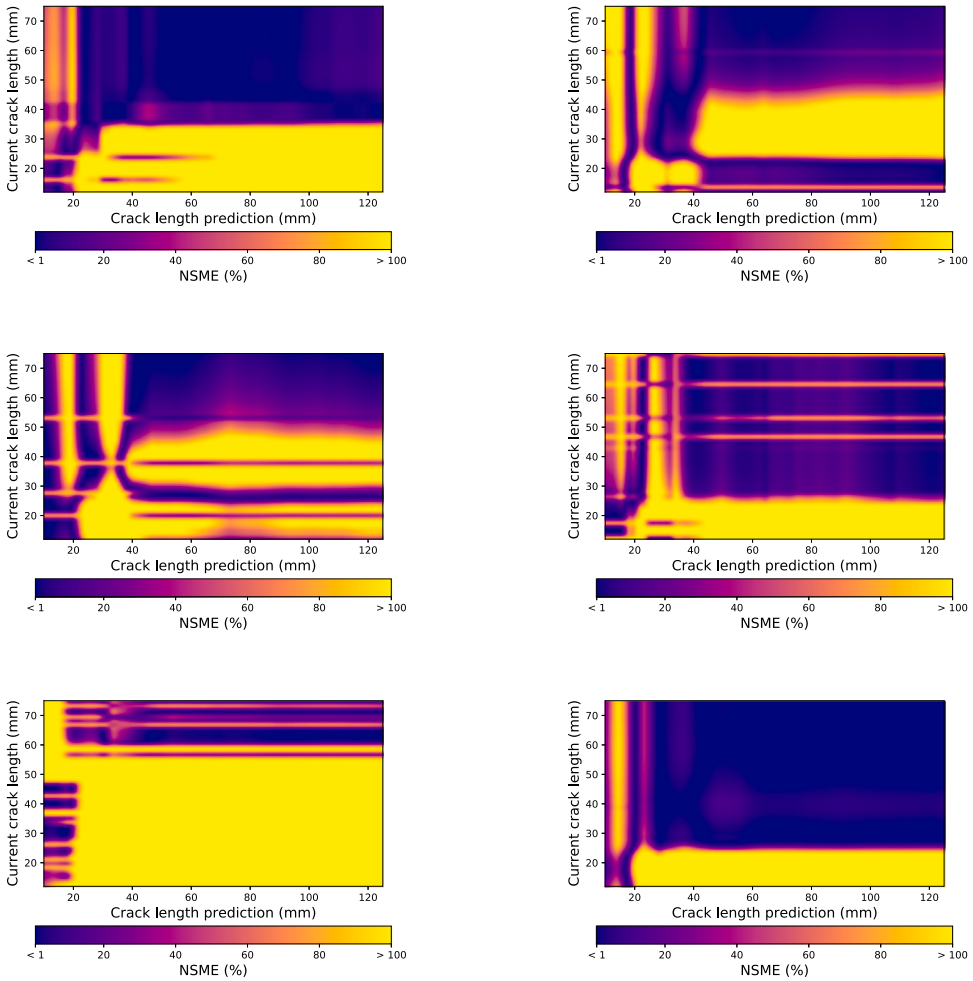

Fig. 8. The resulting normalised mean-square error for the six out-of-population problems regarding the whole damage path, using the fPCA algorithm, as a function of the current crack length and the step for which the prediction is made.

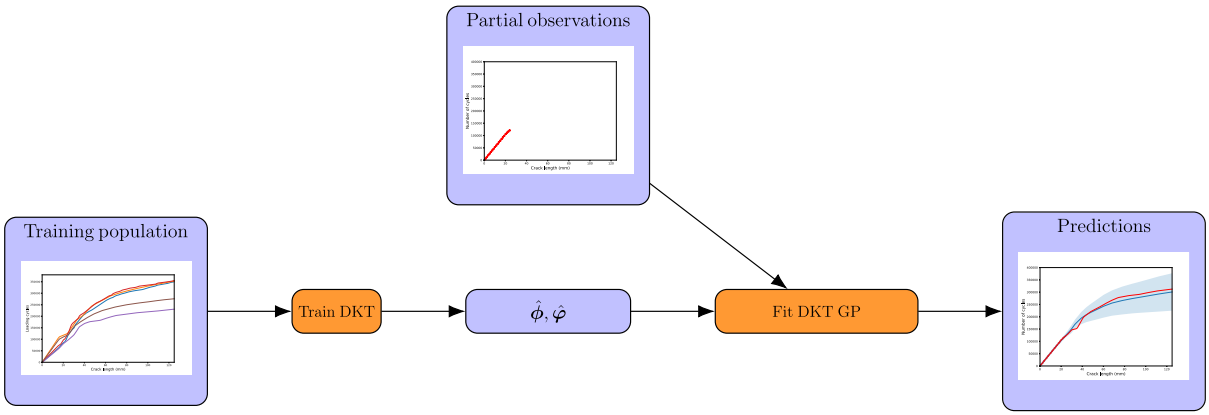

Fig. 9. Schematic representation of the flowchart of applying the deep kernel transfer (DKT) algorithm for damage prognosis.

required. Especially when a large population is considered, it is highly likely that a new structure will have behaviour which is part of the potential behaviours of the existing population or quite similar. Moreover, according to the augmentation scheme followed here, the new samples are generated on the basis of the linear superposition of the behaviours of members of the population. This

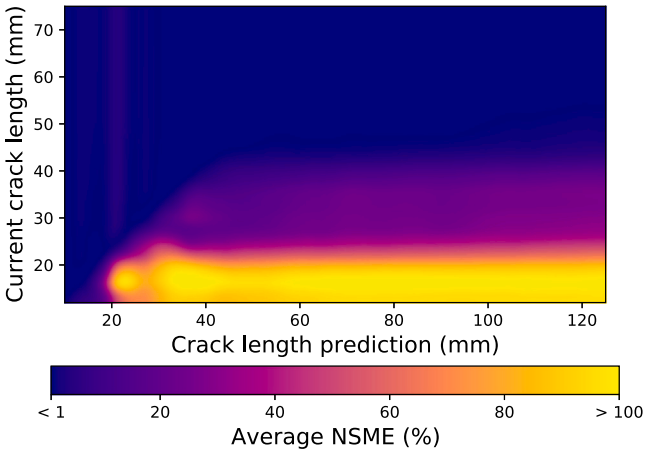

Fig. 10. Average normalised mean-squared error of the predictions of the DKT model for a testing population of 1000 structures. The horizontal axis represents the crack length to which the predictions refer and the vertical axis represents the current crack length.

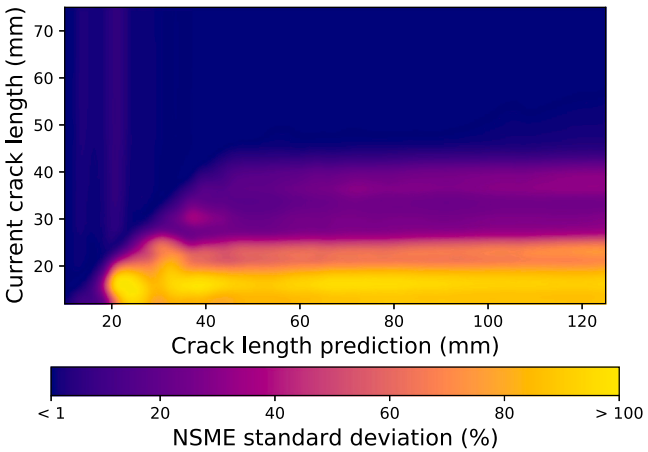

Fig. 11. Standard deviation of the normalised mean-squared error of the predictions of the DKT model for a testing population of 1000 structures. The horizontal axis represents the crack length to which the predictions refer and the vertical axis represents the current crack length.

type of augmentation might as well be physically admissible and, in the specific framework, assists in the creation of a prior belief via the use of a normalising flow.

For the out-of-population problems, similar behaviour is observed. Although the algorithms need more observations to provide accurate predictions compared to the in-population case, after observing more values of the current crack length, the predictions become quite accurate in terms of the NMSE. This behaviour is encouraging regarding the performance of both the fPCA and the DKT algorithms. For both cases, as in the in-population predictions, it is clear that the observations in the initial part of the crack growth region are not quite informative. Observing Fig. 2, it becomes evident that the cracks in all plates evolve quite similarly until they reach a value of approximately 45 mm. This is reflected in the accuracy of the algorithms, which is quite low for values of observed cracks lower than this value. The low accuracy might also be the result of very low standard deviations in this region, making the NMSE quite high, even for low values of actual errors. This behaviour of the presented algorithms is quite natural and reflects their Bayesian nature, i.e. the updating of the predictions according to the information content of the observations, which in many cases may be a desired behaviour.

Another important aspect of the out-of-population predictions is that again, both algorithms achieve a satisfactory level of accuracy after acquiring data corresponding to crack lengths higher than 45 mm. The algorithms seem to adapt slower than in the case of the in-population predictions but this is expected. The increasing accuracy is again encouraging for the two algorithms for different reasons. For the fPCA algorithm, the appropriate principal component scores for the left-out plates may not be part of the prior belief defined by the NF, but there exist values in the hyperplane that correspond to a curve with satisfactory accuracy for the observations. As a result, the HMC algorithm, given enough time, is able to converge to such values and provide appropriate predictions. The DKT algorithm is also performing quite satisfactorily for the out-of-population problems. In this case, the accuracy might be the result of the neural network of the algorithm learning an appropriate transformation of the input variables of the

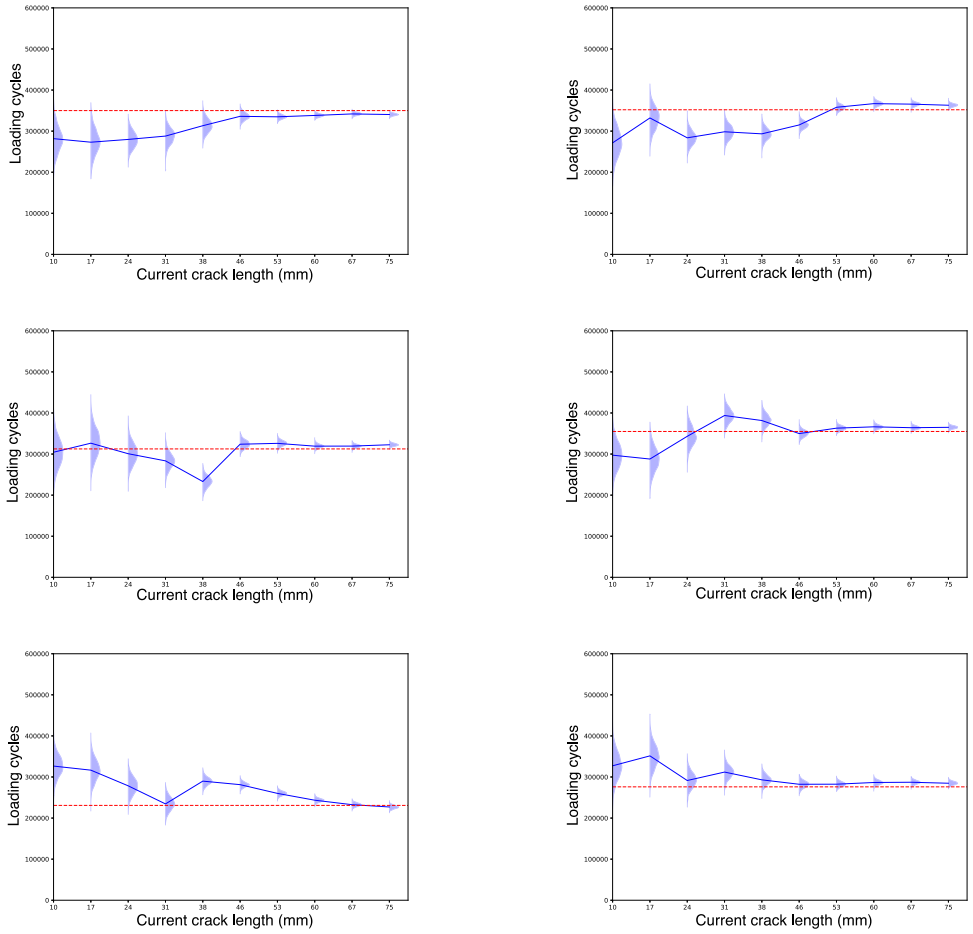

Fig. 12. The PDFs of the estimated total lifetime of the structures using the DKT algorithm, as a function of the current crack length. The mean values of the predictions are shown as the blue curves and the actual total life-time of each plate is shown as the horizontal red line. The numbers of the plates are one to six from right to left and from top to bottom.

GP, so that the mean and covariance functions of the GP resemble the properties of the population. A simple example of such a transformation would be that, given a linear mean function for the GP, the neural network could wrap the input space so that the mean would resemble a logarithmic function, which is a better fit for the current population.

Examining more qualitatively the fitting and the predictions of the algorithms in the Appendix, a quite interesting behaviour is observed. In many cases, the algorithms under- or overestimate the remaining lifetime of the plates. The false estimations happen especially for plates with sudden change in the crack-growth rate. A major example for both algorithms are the fourth and the sixth plates. In both cases there is an underestimation of the total lifetime of the plate using the initial observations. However, for the fourth plate, when the crack length reaches a value of 25 mm, a sudden change in the crack-growth rate is observed. As observations after the change are included in the procedure, the algorithms are able to make more accurate predictions. For the sixth plate, a sudden change is observed twice, and again the algorithms seem to adapt to the observations and to make more accurate predictions. This characteristic of the behaviour of the algorithms might be considered an advantage, because of how well they seem to be informed by the available observations.

An interesting observation regarding the results of the fifth plate in the Appendix is that the fPCA algorithm has high errors for a small amount of available data samples, but the DKT does not have a similar problem with the corresponding testing plate. This difference in behaviour is partially expected. The fPCA algorithm learns a functional subspace represented in a low-dimensional Cartesian space by a normalising flow. It is expected that, when the fifth damage curve is not included, which exhibits significantly quicker failure compared to the rest of the plates, the shaped prior belief might not include coefficients to account for this behaviour, i.e. the point which corresponds to the fifth curve may be outside the domain defined by the normalising flow. In contrast, the DKT algorithm learns correlations between different points of the input space; as a result, it can deal with such differences in the crack-length evolution.

In both cases, a population-informed predictive framework is achieved. Part of the utility of the two algorithms is that they both attempt to learn the underlying physics in a physics-blind scheme. What this means is that the analysts do not need to impose

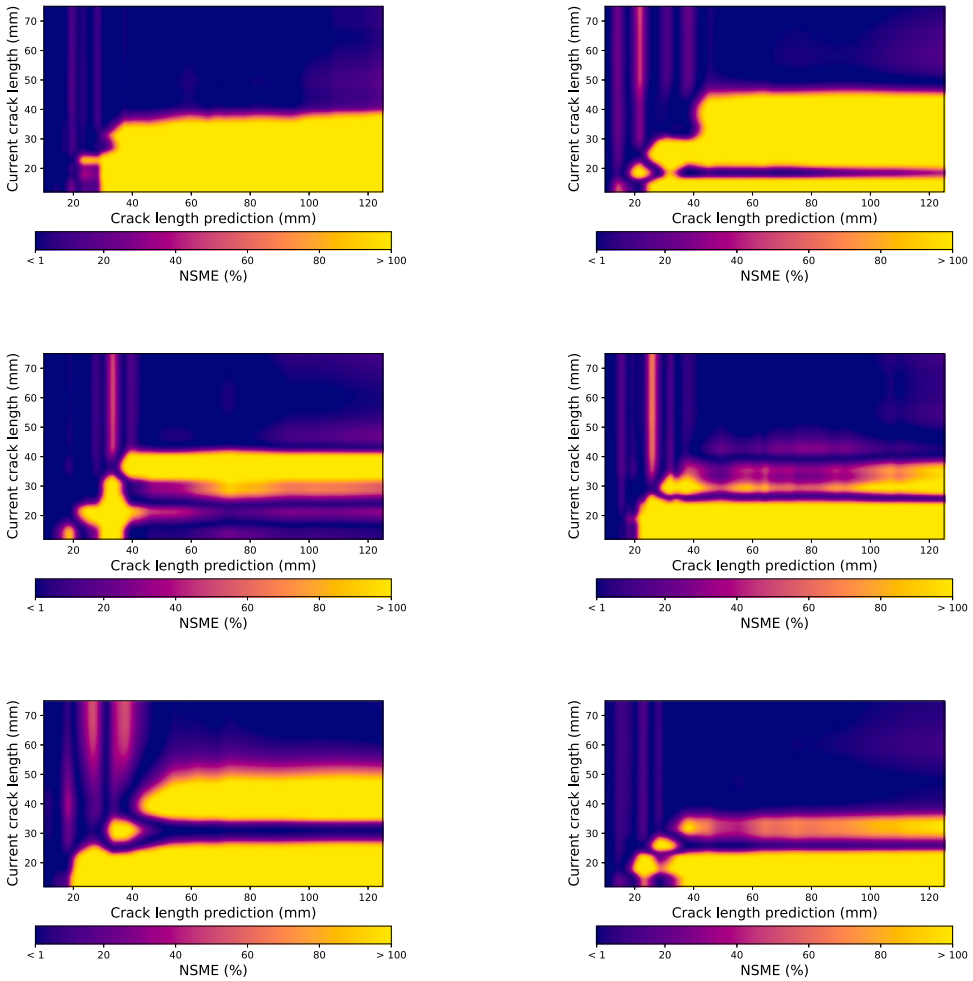

Fig. 13. The resulting normalised mean-square error for the six out-of-population problems regarding the whole damage path, using the DKT algorithm, as a function of the current crack length and the step for which the prediction is made.

any knowledge into the models. This attribute could indicate that the models could perform in a population-based framework of modelling of a wide variety of phenomena and systems, and not only in the domain of structural dynamics and damage prognosis. Although the two algorithms were applied to a nominally-identical population, in future work, new algorithms might be developed incorporating methodologies similar to the ones presented in [2–5]. Exploiting metrics which could quantify the similarity of structures, the current work could be extended to heterogeneous populations of systems.

7. Conclusions

The current work presents two methods to perform population-based structural health monitoring. The first method is based on defining a functional subspace which comprises the functions that explain the behaviour of members of a population of structures. The second method is a meta-learning algorithm, according to which a neural network is employed as a transformation function for the inputs to a Gaussian process. The functionality of the neural network is to transform the features, so that the GP mean and covariance functions represent appropriately the actual underlying mean and covariance functions of the population. The problem, which is studied here, is that of damage prognosis. Such a problem is inherently a problem of imbalanced data, because one can only have access to the data in the beginning of the evolution of the phenomenon.

A difference between the current work and other damage-prognosis methods is that the input to the inference algorithm is the crack length and the output is the number of cycles required to achieve such a value of crack length. Making this change in the problem formulation allows the application of the proposed algorithms. The inversion of the input and output variables allows a proper definition of the functions of the behaviour of the structures under evolving damage, because this way the support of the damage-evolution curves is common for every structure, i.e. for every value of the input space (crack length) there is a value in the output space (number of cycles), which is not the case for the opposite consideration. By moving the uncertainty to the output, this

consideration could potentially allow an easier implementation of other algorithms as well, since the exact number of loading cycles between each crack-length measurements may be quite inaccurate, because of varying environmental end operational conditions, but the crack length measurements can be quite accurate.

To deal with the specific problem, the two methods are applied and evaluated according to their performance on an experimental crack-growth dataset and for two categories of testing populations. The first category is an in-population testing dataset, which includes testing structures which come from the same population of structures as the training structures. The algorithms are quite accurate in this case, especially as more observations from the testing structures are included in the inference process. The second category considers out-of-population testing structures. More specifically the testing structures are plates from the experiments, which were left out of the training population. Because the actual plate population is small (six plates) and they have quite different behaviour, the left-out plates are not considered part of the population. In this case, the algorithms also exhibit satisfactory behaviour. Given enough observations from the testing plates, the predictions become quite accurate in terms of predicting the remaining damage path.

Concluding, the proposed methods appear to be learning part of the physics of the evolution of damage in the available population of structures. A big advantage of the methods is that they do not need to be informed by the analyst about the underlying physics and they learn exclusively from data. Moreover, they clearly exhibit a Bayesian behaviour, i.e. they are initially informed by some prior belief by being trained according to the training population, and, as more data become available from the structures of the testing population, their predictions adapt to the specific behaviour of the structures of the testing population. Therefore, the two proposed algorithms could be considered as a potential solution to a plethora of population-based modelling problems and they motivate a functional and population-informed way of thinking about problems, instead of the traditional single-problem (or single-structure) approach.

CRediT authorship contribution statement

G. Tsialiamanis: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **C. Sbarufatti:** Data curation, Investigation, Resources, Supervision, Writing – original draft, Writing – review & editing. **N. Dervilis:** Conceptualization, Investigation, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing. **K. Worden:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing – original draft, Writing – review & editing.

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 authors do not have permission to share data.

Acknowledgments

G.T. and K.W would like to gratefully acknowledge the support of the UK Engineering and Physical Sciences Research Council (EPSRC) via grant references EP/W005816/1. For the purpose of open access, the authors has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

Appendix

Qualitative fPCA results

See Figs. 14–19.

Qualitative DKT results

See Figs. 20–25.

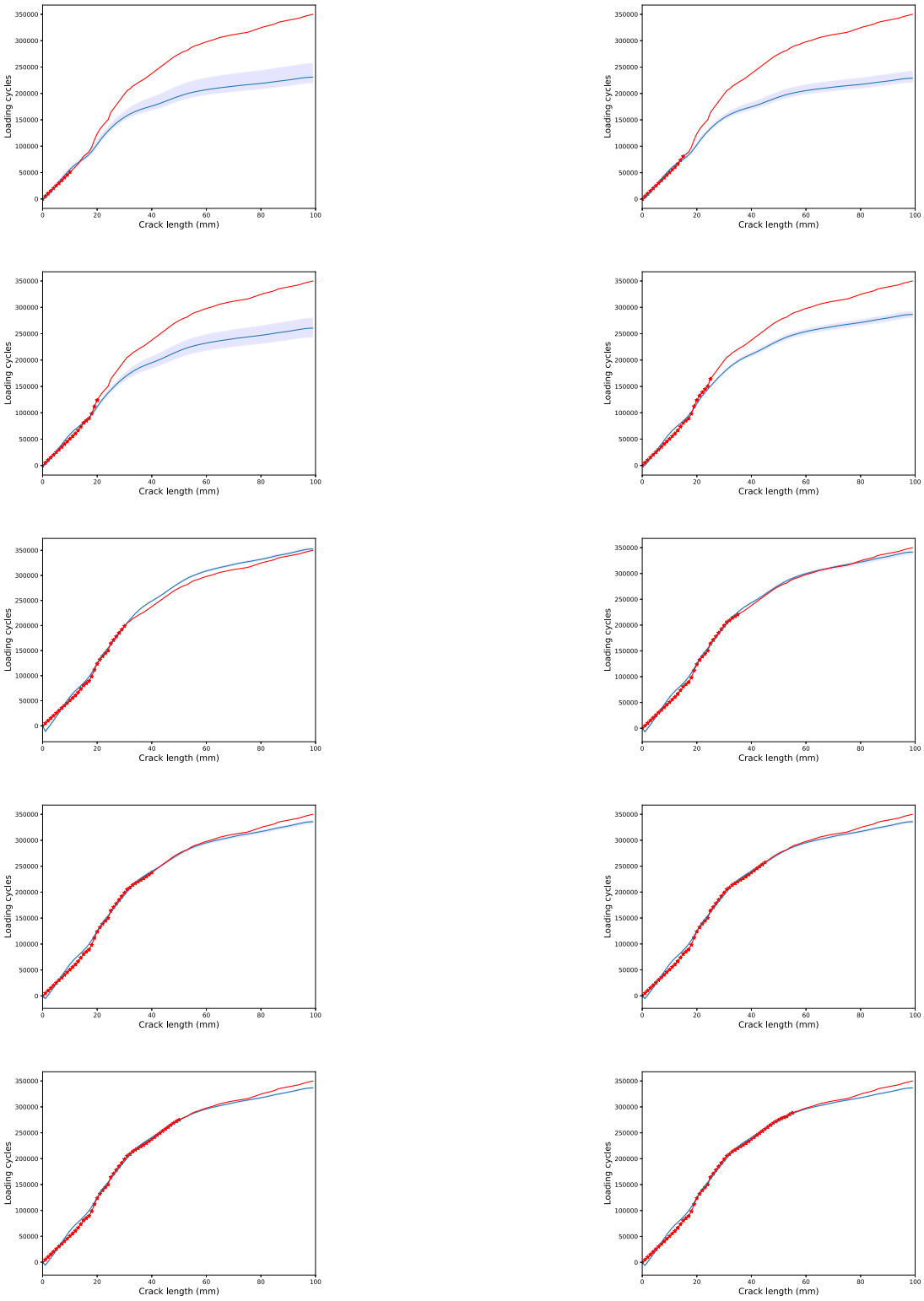

Fig. 14. Mean value of the predictions of the fPCA algorithm (blue curve), minimum–maximum confidence interval of the predictions (blue-shaded area), the true underlying damage path of the first testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

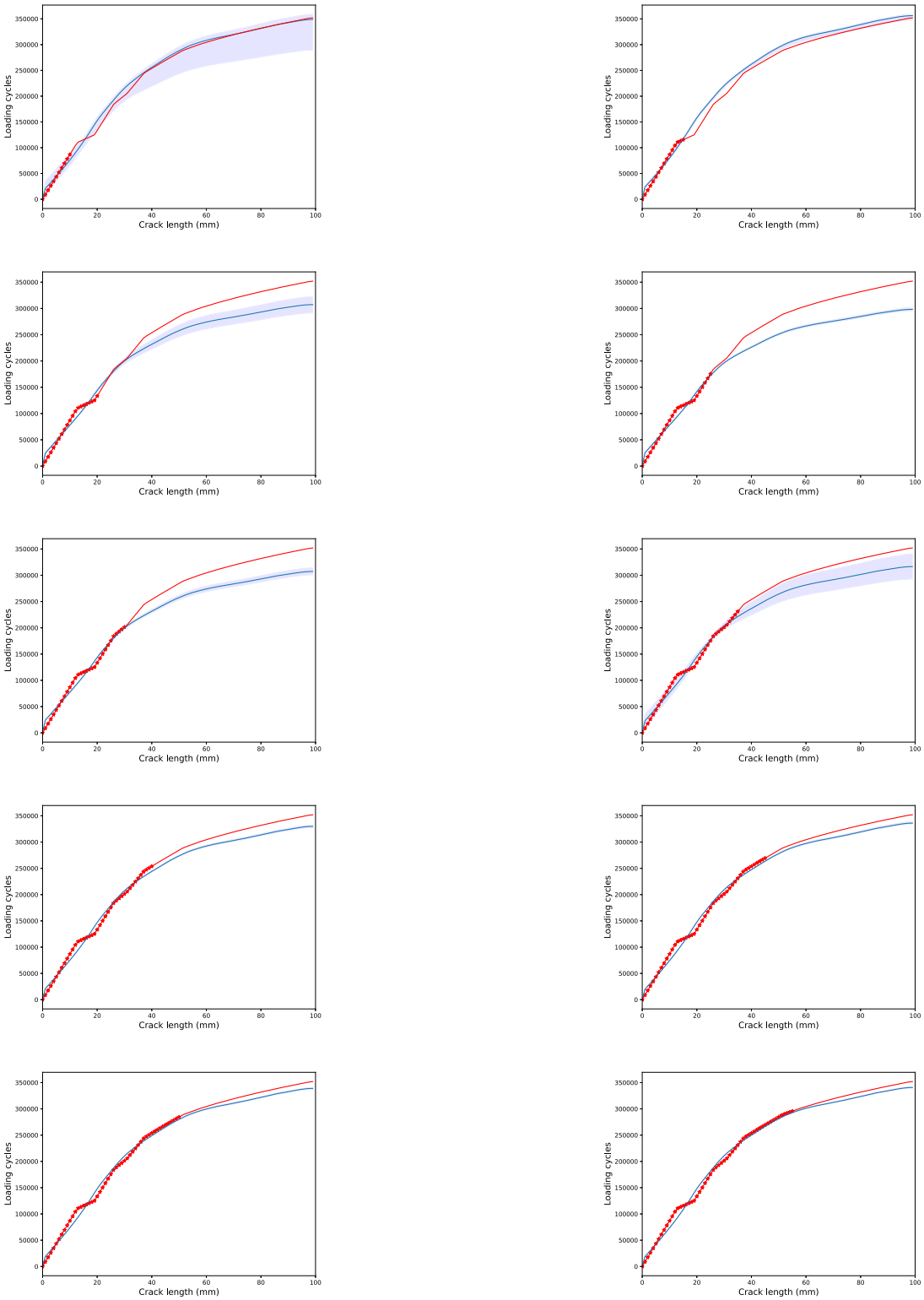

Fig. 15. Mean value of the predictions of the fPCA algorithm (blue curve), minimum-maximum confidence interval of the predictions (blue-shaded area), the true underlying damage path of the second testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

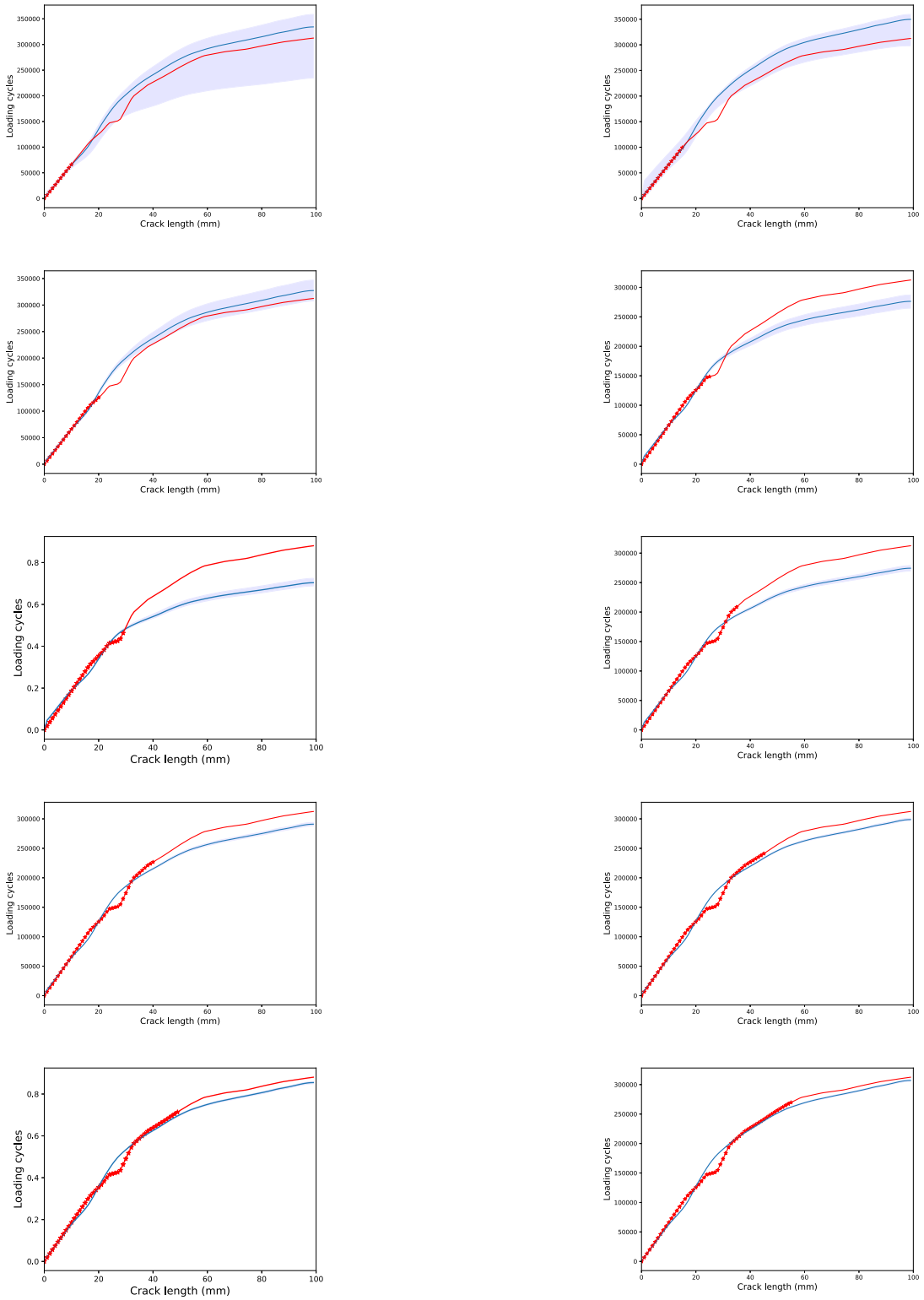

Fig. 16. Mean value of the predictions of the fPCA algorithm (blue curve), minimum-maximum confidence interval of the predictions (blue-shaded area), the true underlying damage path of the third testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

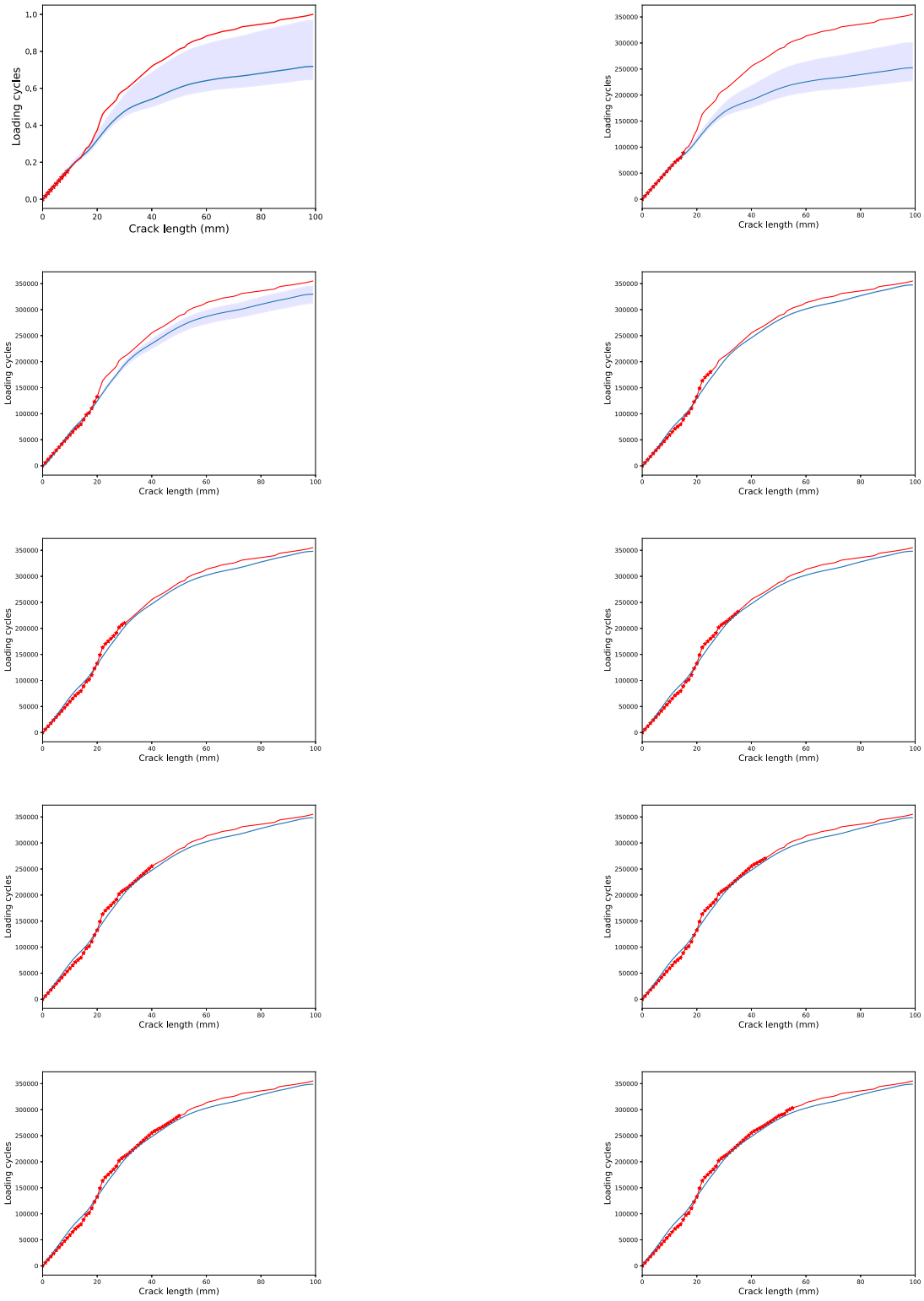

Fig. 17. Mean value of the predictions of the fPCA algorithm (blue curve), minimum-maximum confidence interval of the predictions (blue-shaded area), the true underlying damage path of the fourth testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

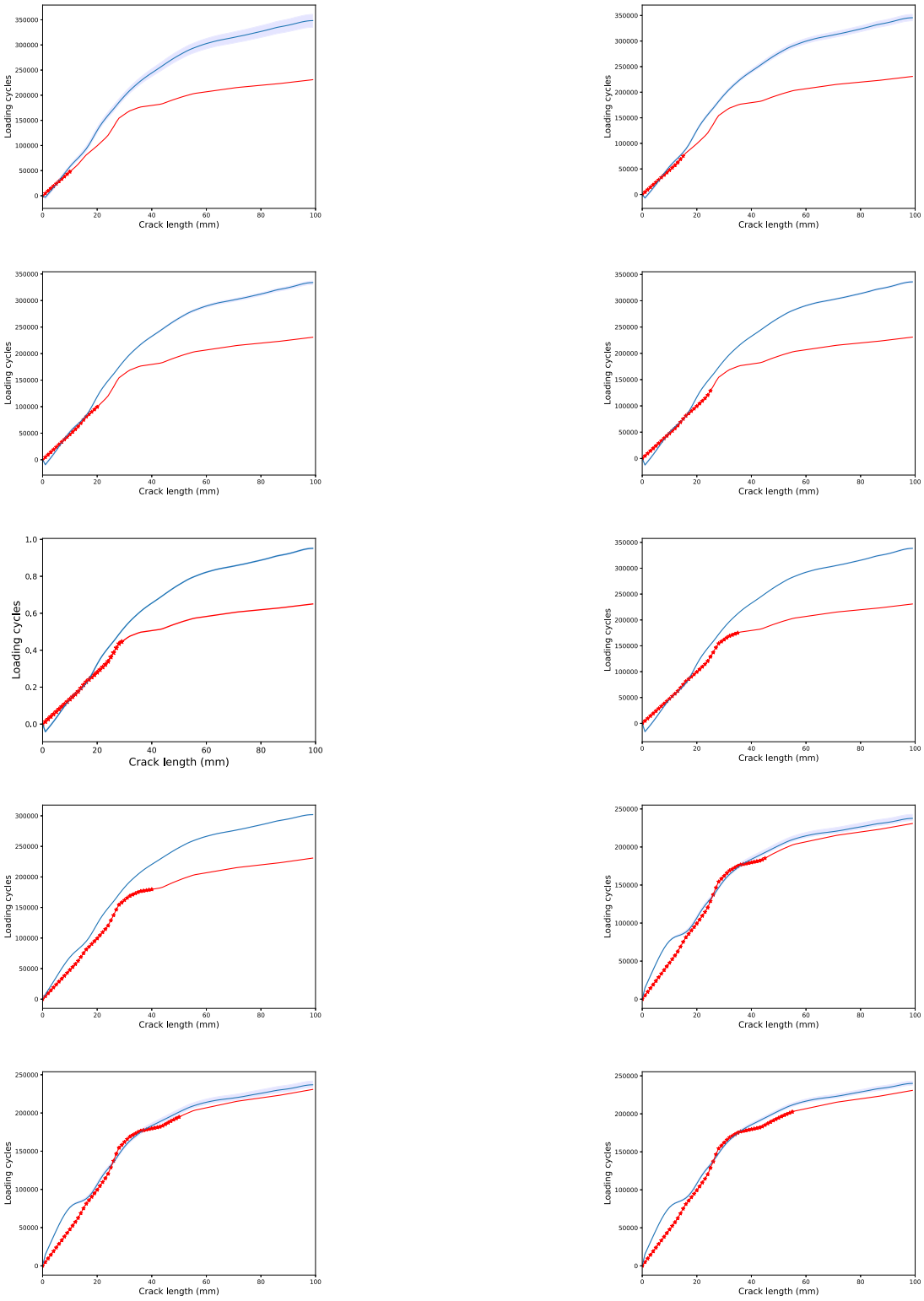

Fig. 18. Mean value of the predictions of the fPCA algorithm (blue curve), minimum-maximum confidence interval of the predictions (blue-shaded area), the true underlying damage path of the fifth testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

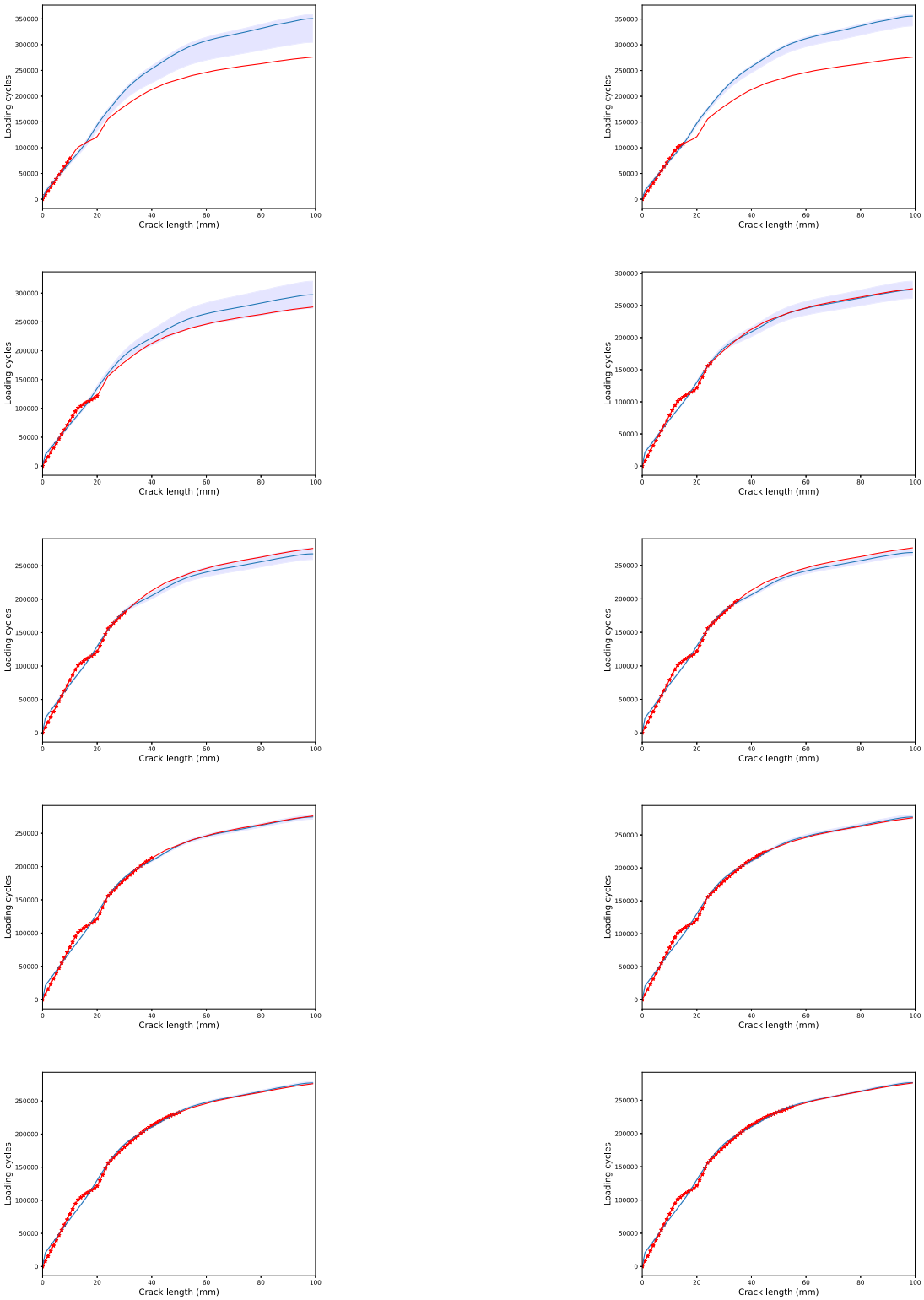

Fig. 19. Mean value of the predictions of the fPCA algorithm (blue curve), minimum-maximum confidence interval of the predictions (blue-shaded area), the true underlying damage path of the sixth testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

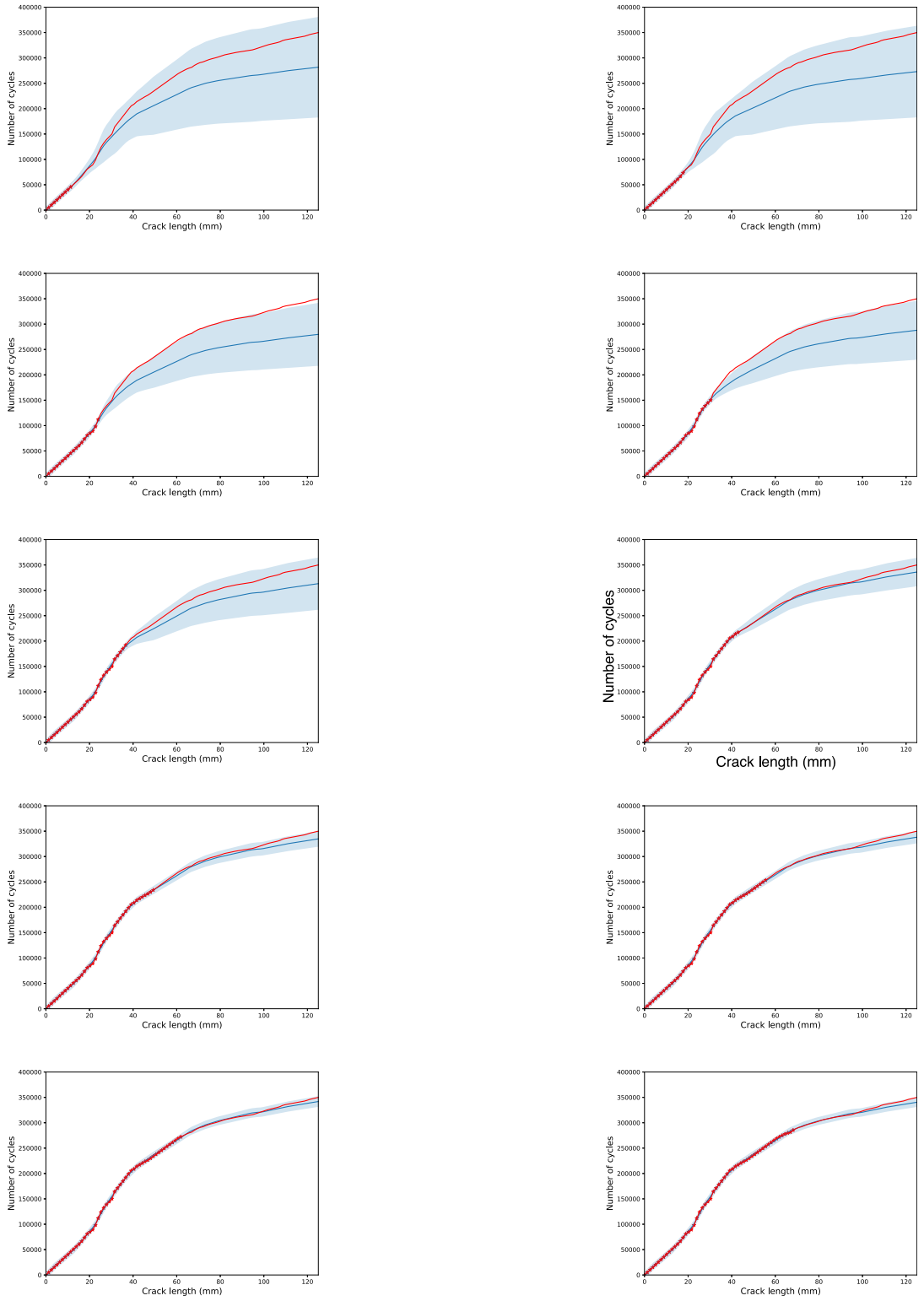

Fig. 20. Mean value of the predictions of the DKT algorithm (blue curve), ±3 standard deviations confidence interval of the predictions (blue-shaded area), the true underlying damage path of the first testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

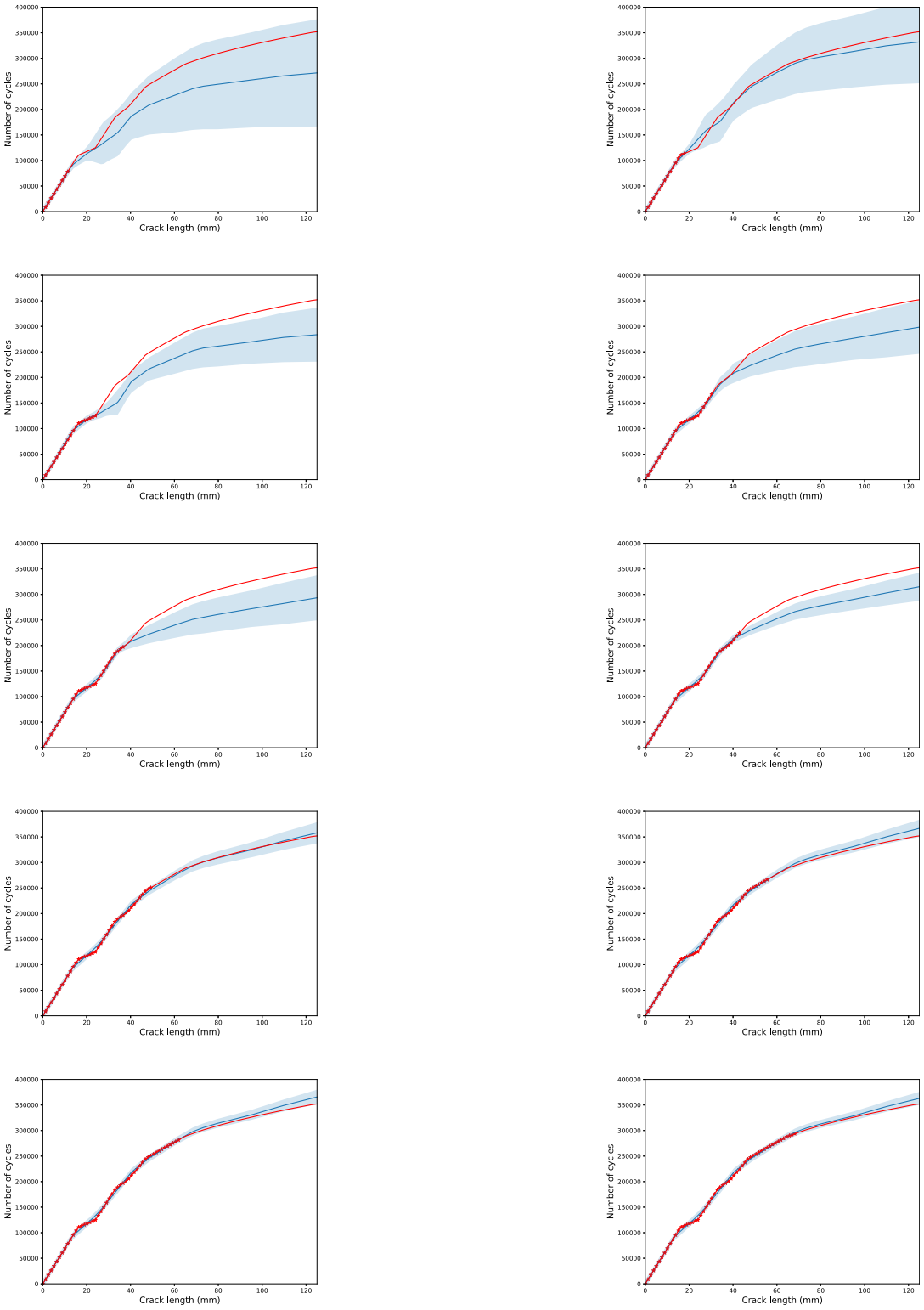

Fig. 21. Mean value of the predictions of the DKT algorithm (blue curve), ± 3 standard deviations confidence interval of the predictions (blue-shaded area), the true underlying damage path of the second testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

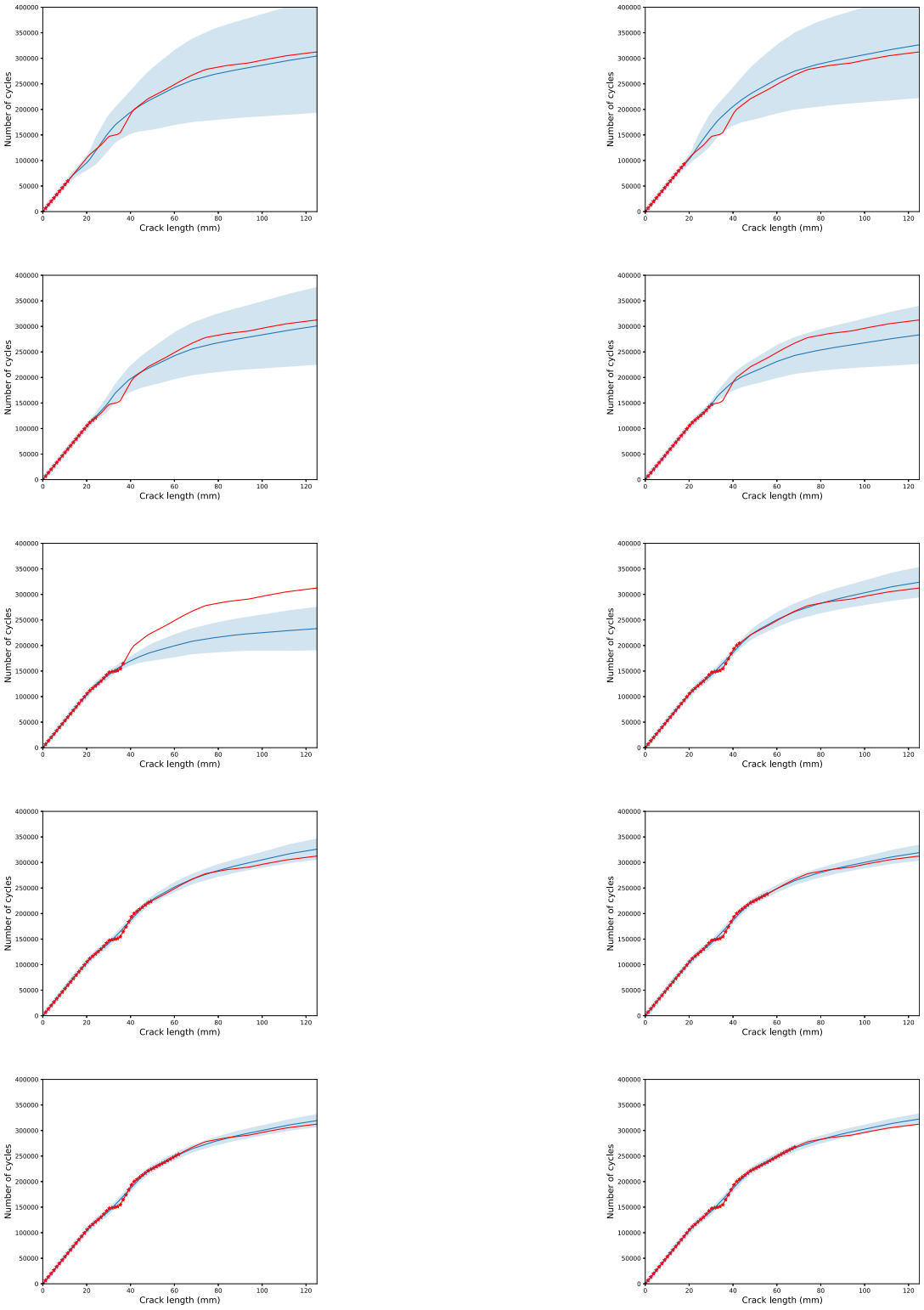

Fig. 22. Mean value of the predictions of the DKT algorithm (blue curve), ± 3 standard deviations confidence interval of the predictions (blue-shaded area), the true underlying damage path of the third testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

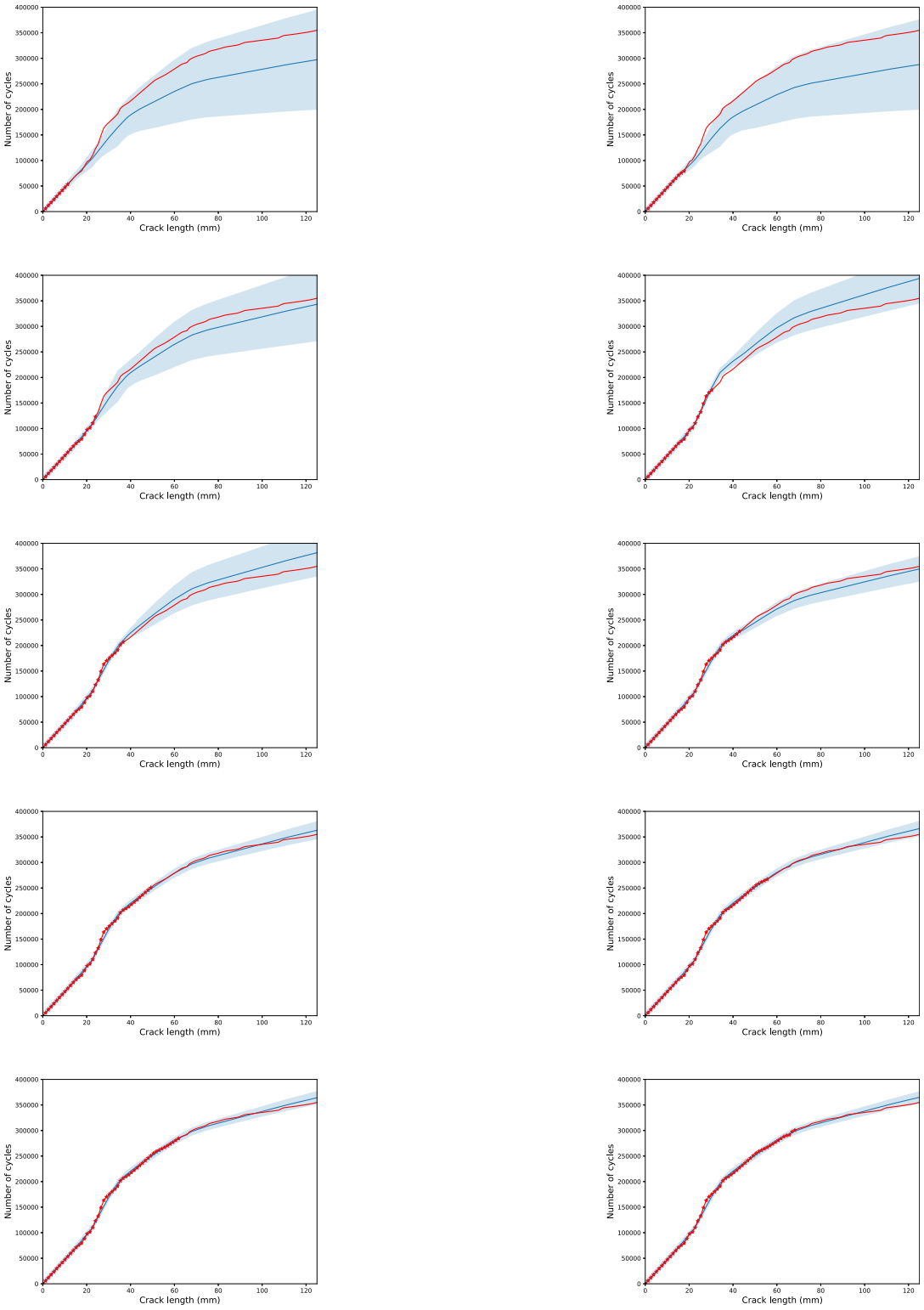

Fig. 23. Mean value of the predictions of the DKT algorithm (blue curve), ± 3 standard deviations confidence interval of the predictions (blue-shaded area), the true underlying damage path of the fourth testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

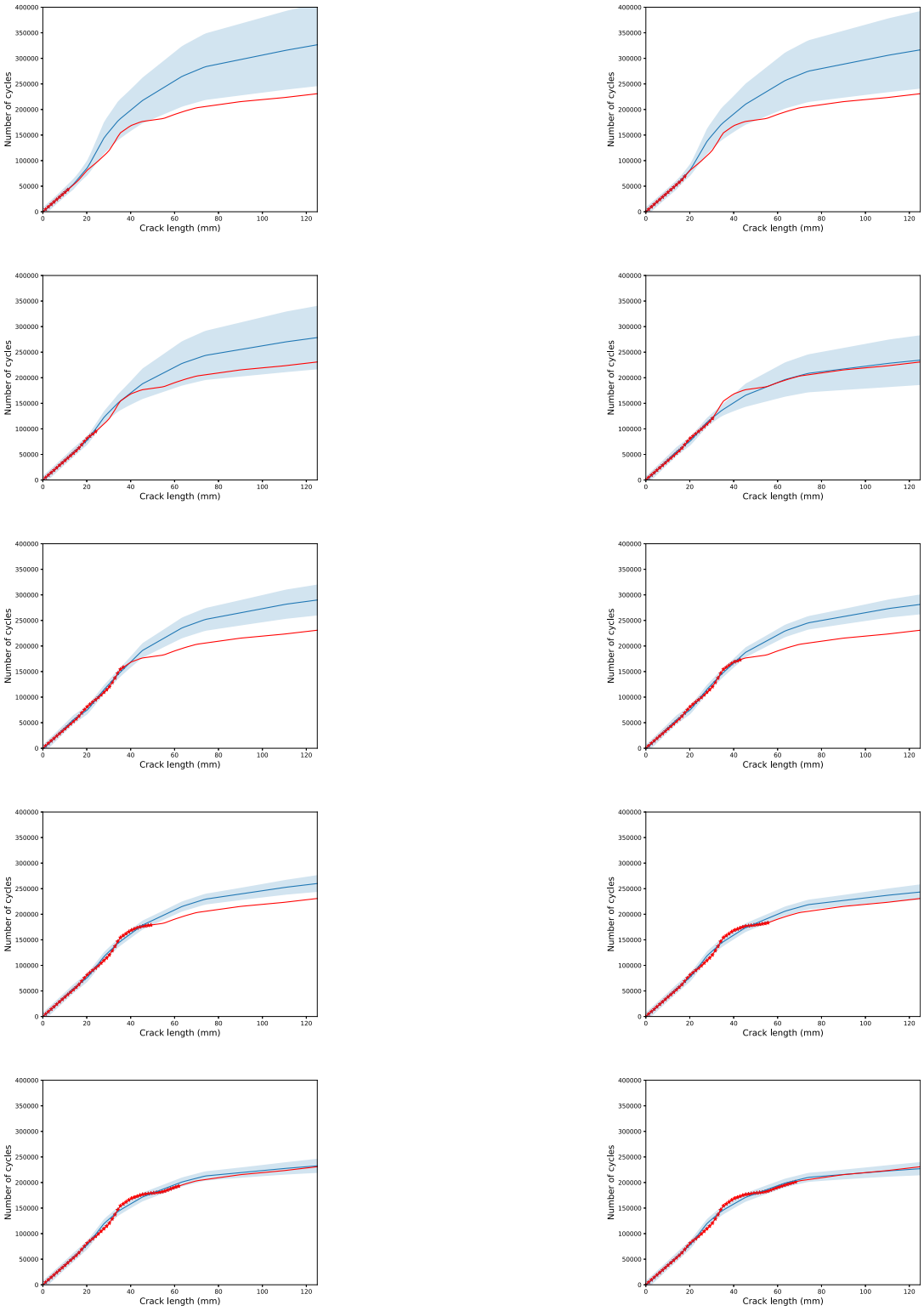

Fig. 24. Mean value of the predictions of the DKT algorithm (blue curve), ± 3 standard deviations confidence interval of the predictions (blue-shaded area), the true underlying damage path of the fifth testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

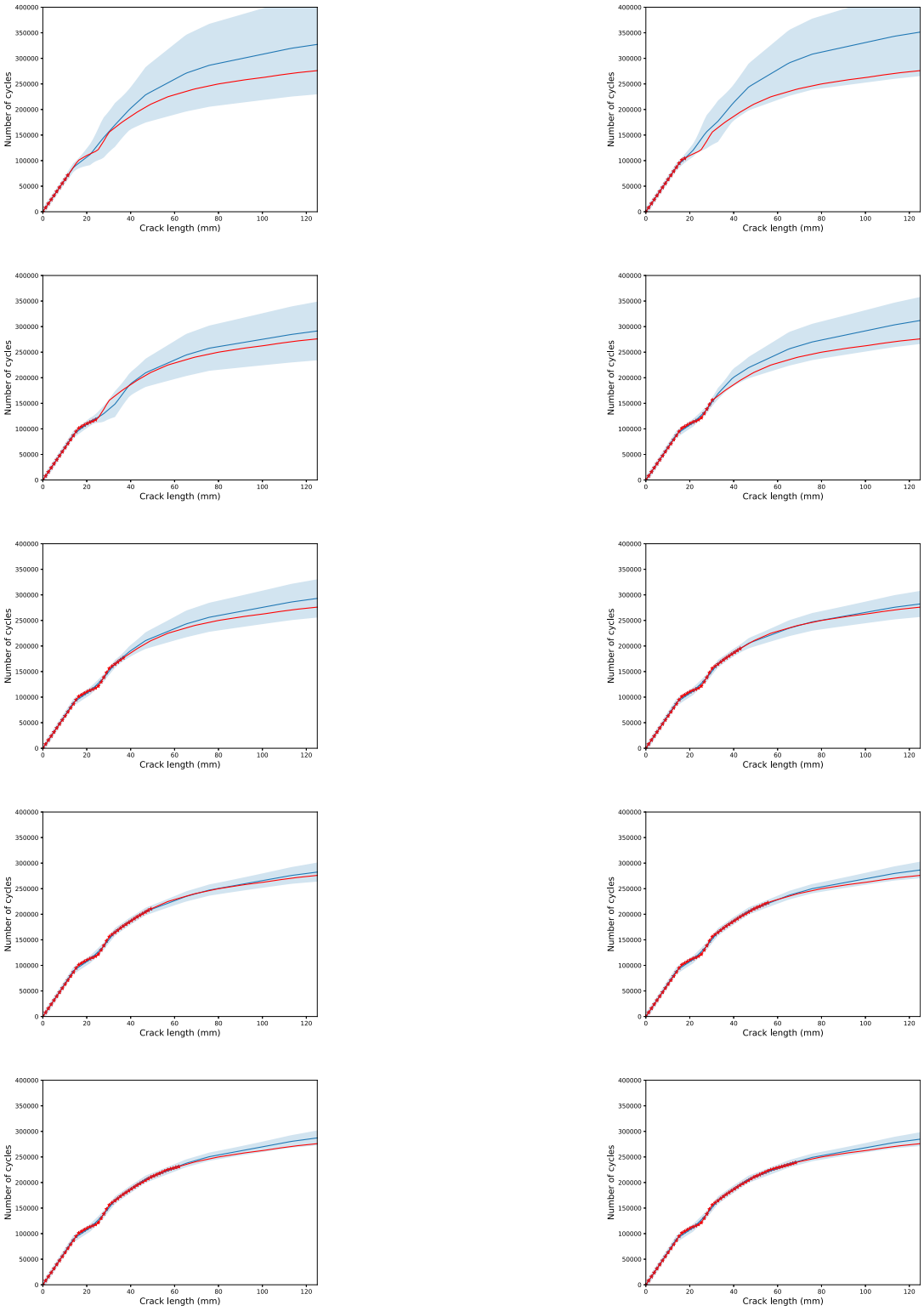

Fig. 25. Mean value of the predictions of the DKT algorithm (blue curve), ±3 standard deviations confidence interval of the predictions (blue-shaded area), the true underlying damage path of the sixth testing plate (red curve), and the observations for different stages of the damage evolution (red stars). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

References

[1] L.A. Bull, P.A. Gardner, J. Gosliga, T.J. Rogers, N. Dervilis, E.J. Cross, E. Papatheou, A.E. Maguire, C. Campos, K. Worden, Foundations of population-based SHM, Part I: Homogeneous populations and forms, Mech. Syst. Signal Process. 148 (2021) 107141.
[2] J. Gosliga, P.A. Gardner, L.A. Bull, N. Dervilis, K. Worden, Foundations of population-based SHM, Part II: Heterogeneous populations–Graphs, networks, and communities, Mech. Syst. Signal Process. 148 (2021) 107144.
[3] P. Gardner, L.A. Bull, J. Gosliga, N. Dervilis, K. Worden, Foundations of population-based SHM, Part III: Heterogeneous populations–mapping and transfer, Mech. Syst. Signal Process. 149 (2021) 107142.
[4] G. Tsialiamanis, C. Mylonas, E. Chatzi, N. Dervilis, D.J. Wagg, K. Worden, Foundations of population-based SHM, Part IV: The geometry of spaces of structures and their feature spaces, Mech. Syst. Signal Process. 157 (2021) 107692.
[5] D.S. Brennan, J. Gosliga, C.T. Wickramarachchi, A. Bunce, D. Hester, E.J. Cross, K. Worden, Foundations of population-based SHM, Part V: Networks and databases, Mech. Syst. Signal Process. (2023) (submitted for publication).
[6] C.R. Farrar, K. Worden, Structural Health Monitoring: A Machine Learning Perspective, John Wiley and Sons, 2011.
[7] K. Worden, G. Manson, D. Allman, Experimental validation of a structural health monitoring methodology: Part I. Novelty detection on a laboratory structure, J. Sound Vib. 259 (2) (2003) 323–343.
[8] G. Manson, K. Worden, D. Allman, Experimental validation of a structural health monitoring methodology: Part II. Novelty detection on a Gnat aircraft, J. Sound Vib. 259 (2) (2003) 345–363.
[9] G. Manson, K. Worden, D. Allman, Experimental validation of a structural health monitoring methodology: Part III. Damage location on an aircraft wing, J. Sound Vib. 259 (2) (2003) 365–385.
[10] Plato, Protagoras and Meno, Penguin Classics, 2005.
[11] Plato, The Republick, Penguin Classics, 2007.
[12] C.E. Rasmussen, Gaussian processes in machine learning, in: Summer School on Machine Learning, Springer, 2003, pp. 63–71.
[13] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Inc, New York, NY, USA, 1995.
[14] C.M. Bishop, Pattern Recognition and Machine Learning, Springer-Verlag, 2006.
[15] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, Vol. 1, No. 2, MIT press Cambridge, 2016.
[16] P.W. Battaglia, J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al., Relational inductive biases, deep learning, and graph networks, 2018, arXiv preprint arXiv:1806.01261.
[17] C.R. Farrar, N.A.J. Lieven, Damage prognosis: the future of structural health monitoring, Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. 365 (1851) (2007) 623–632.
[18] M. Corbetta, C. Sbarufatti, M. Giglio, M.D. Todd, Optimization of nonlinear, non-Gaussian Bayesian filtering for diagnosis and prognosis of monotonic degradation processes, Mech. Syst. Signal Process. 104 (2018) 305–322.
[19] P. Gardner, L.A. Bull, N. Dervilis, K. Worden, Overcoming the problem of repair in structural health monitoring: Metric-informed transfer learning, J. Sound Vib. 510 (2021) 116245.
[20] P. Paris, F. Erdogan, A critical analysis of crack propagation laws, 1963.
[21] K. Agathos, E. Chatzi, S.P.A. Bordas, Multiple crack detection in 3D using a stable XFEM and global optimization, Comput. Mech. 62 (2018) 835–852.
[22] K.T.P. Nguyen, M. Fouladirad, A. Grall, Model selection for degradation modeling and prognosis with health monitoring data, Reliab. Eng. Syst. Saf. 169 (2018) 105–116.
[23] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533–536.
[24] J.O. Ramsay, B.W. Silverman, Principal components analysis for functional data, Funct. Data Anal. (2005) 147–172.
[25] J. Vanschoren, Meta-learning: A survey, 2018, arXiv preprint arXiv:1810.03548.
[26] M. Patacchiola, J. Turner, E.J. Crowley, M. O'Boyle, A.J. Storkey, Bayesian meta-learning for the few-shot setting via deep kernels, in: Advances in Neural Information Processing Systems, Vol. 33, 2020, pp. 16108–16118.
[27] V. Vapnik, The Nature of Statistical Learning Theory, Springer Science & Business Media, 1999.
[28] S. Duane, A.D. Kennedy, B.J. Pendleton, D. Roweth, Hybrid Monte Carlo, Phys. Lett. B 195 (2) (1987) 216–222.
[29] R.M. Neal, Bayesian Learning for Neural Networks, Vol. 118, Springer Science & Business Media, 2012.
[30] T. Hospedales, A. Antoniou, P. Micaelli, A. Storkey, Meta-learning in neural networks: A survey, 2020, arXiv preprint arXiv:2004.05439.
[31] S.J. Pan, Q. Yang, A survey on transfer learning, IEEE Trans. Knowl. Data Eng. (2010).
[32] S. Ruder, An overview of multi-task learning in deep neural networks, 2017, arXiv preprint arXiv:1706.05098.
[33] K.-J. Bathe, Finite Element Procedures, Klaus-Jurgen Bathe, 2006.
[34] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 1126–1135.
[35] M. Andrychowicz, M. Denil, S. Gomez, M.W. Hoffman, D. Pfau, T. Schaul, B. Shillingford, N. De Freitas, Learning to learn by gradient descent by gradient descent, in: Advances in Neural Information Processing Systems, Vol. 29, 2016.
[36] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y.W. Teh, D. Rezende, S.M.A. Eslami, Conditional neural processes, in: International Conference on Machine Learning, PMLR, 2018, pp. 1704–1713.
[37] Y. Wang, Q. Yao, J.T. Kwok, L.M. Ni, Generalizing from a few examples: A survey on few-shot learning, ACM Comput. Surv. (CSUR) 53 (3) (2020) 1–34.
[38] E.J. Cross, T.J. Rogers, T.J. Gibbons, Grey-box modelling for structural health monitoring: physical constraints on machine learning algorithms, Struct. Health Monit. 2019 (2019).
[39] M. Corbetta, C. Sbarufatti, A. Manes, M. Giglio, On dynamic state-space models for fatigue-induced structural degradation, Int. J. Fatigue 61 (2014) 202–219.
[40] C. Sbarufatti, A. Manes, M. Giglio, Performance optimization of a diagnostic system based upon a simulated strain field for fatigue damage characterization, Mech. Syst. Signal Process. 40 (2) (2013) 667–690.
[41] D. Rezende, S. Mohamed, Variational inference with normalizing flows, in: International Conference on Machine Learning, PMLR, 2015, pp. 1530–1538.
[42] G. Papamakarios, E. Nalisnick, D.J. Rezende, S. Mohamed, B. Lakshminarayanan, Normalizing flows for probabilistic modeling and inference, J. Mach. Learn. Res. 22 (1) (2021) 2617–2680.
[43] I. Kobyzev, S.J.D. Prince, M.A. Brubaker, Normalizing flows: An introduction and review of current methods, IEEE Trans. Pattern Anal. Mach. Intell. 43 (11) (2020) 3964–3979.
[44] L. Dinh, J. Sohl-Dickstein, S. Bengio, Density estimation using real nvp, 2016, arXiv preprint arXiv:1605.08803.
[45] M.D. Hoffman, A. Gelman, et al., The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo, J. Mach. Learn. Res. 15 (1) (2014) 1593–1623.
[46] Adam D. Cobb, Brian Jalaian, Scaling Hamiltonian Monte Carlo inference for Bayesian neural networks with symmetric splitting, in: Uncertainty in Artificial Intelligence, 2021.
[47] L. Yang, D. Zhang, G.E. Karniadakis, Physics-informed generative adversarial networks for stochastic differential equations, SIAM J. Sci. Comput. 42 (1) (2020) A292–A317.