Through-Life Monitoring of Resource-constrained Systems and Fleets

Felipe Montana[1], Will Jacobs[1], Oscar Mendoza[1], Visakan Kadirkamanathan[1], Nima Ameri[2], Philip Naylor[2], and Andy Mills[1]

[1] *University of Sheffield, Sheffield, S1 3JD, UK*
f.montana-gonzalez@sheffield.ac.uk
w.jacobs@sheffield.ac.uk
o.e.mendoza@sheffield.ac.uk
visakan@sheffield.ac.uk
a.r.mills@sheffield.ac.uk

[2] *Rolls-Royce Plc, London, N1 9FX, UK*
nima.ameri@rolls-royce.com
philip.naylor@rolls-royce.com

ABSTRACT

A Digital Twin (DT) is a representation of a physical system that provides information to make decisions that add economic, social or commercial value. DTs are widely used for prognostics and anomaly detection by continuously comparing measured system behaviour with the DT predictions to identify deviations and estimate degradation. The behaviour of a physical system changes over time; a DT must therefore be continually updated with data from the physical system to reflect its changing behaviour. In this paper, we consider a DT of a complex, non-linear and dynamic system subject to slow nominal degradation, disturbances and the risk of anomalies. The DT runs on a resource-constrained system, making updating non-trivial due to limitations in computation, storage, and data transfer bandwidth. Consequently, only a subset of the generated data can be retained or transmitted, making data prioritisation essential. Data must be evaluated online in order to select the most relevant subset with which to perform the update. DT updating must address the continual learning challenge of adapting to new system behaviours, such as the response to previously unseen operating conditions, while retaining knowledge of previously observed behaviours. This paper presents a framework for updating a data-driven DT of a resource-constrained system. The proposed solution consists of: (1) an on-board, lightweight DT that enables the prioritisation and parsimonious transfer of data generated by the physical system; and (2) an off-board system for robust DT updating that enables the reliable detection of anomalous behaviours across the asset's lifetime. The framework allows the DT to accurately replicate the behaviour of the system throughout its life, improve sensitivity to anomaly detection, and reduce the risk of forgetting previous system behaviours after updating the DT. An in-service gas turbine engine case study is used for demonstration.

1. INTRODUCTION

A Digital Twin (DT) is an emerging technology that has recently gained attention due to the rapid development of simulations, data acquisition and data communication that trigger interaction between the physical and virtual spaces (Tao, Zhang, Liu, & Nee, 2018). In (Glaessgen & Stargel, 2012), a DT is defined as a simulation of an as-built system, e.g., a vehicle or factory, that accurately represents its corresponding twin. We extend this definition to include the continuous update of a simulation, reflecting the 'as-operated' behaviours and, as stated in (*The DTHub*, 2026), enabling decisions that will affect the physical asset. It is commonly recognised that a DT needs to integrate data collected from the system's sensors, environment, historic maintenance data, and available system knowledge to reflect any changes to the physical asset. The information provided by the DT facilitates the processes of making decisions that add economic, social, or commercial value to stakeholders.

The DT paradigm has been used in a variety of applications and sectors, including product design and factory optimisation (Qi & Tao, 2018), prediction of aircraft structural life (Tuegel, Ingraffea, Eason, & Spottswood, 2011), monitor-

---

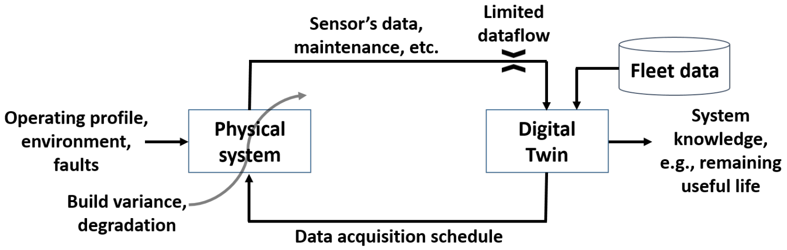

Figure 1. Data flow in a Digital Twin. A DT must be routinely updated, with data recorded from the physical system, to reflect changes due to degradation, build variance, etc. Anomalous data and constraints in data collection are some of the challenges faced when updating the DT. The DT provides knowledge about the physical system and the means to simulate the system under different conditions.

ing of automotive braking systems (Magargle et al., 2017), among other applications. Another example is the management and monitoring of a fleet of assets. In such cases, information based on the fleet distribution is inadequate to assess individual systems due to the variability in usage, manufacturing and material properties (C. Li, Mahadevan, Ling, Choze, & Wang, 2017). Hence, the availability of individually tailored models or DTs is desirable to accurately predict future performance against requirements and detect deviation from the current system behaviour driven by emerging faults.

Changes in an individual physical system are expected due to normal degradation, mechanical modifications, etc., and this results in a drift between the twin and the real system. In addition to this drift, twin inaccuracy may be due to system operation in previously unseen operating conditions. Both scenarios motivate the model's update using measurement data acquired from the physical system and thus maximise the model's representative power.

In our prior work, we have developed a DT that has been demonstrated to detect real-world anomalies in a resource-constrained computing environment mounted onto a gas turbine engine (Hartwell et al., 2024). Our DT generates a high anomaly score for a data point when its uncertainty-weighted prediction error is high. In the previous work, we highlighted the need for adaptation to novel operation conditions and engine aging. Without an update of the DT, the anomaly detection performance decreases in the long term. This paper explores the DT adaptation and improvement of anomaly detection performance throughout asset life.

The process to update a model is typically many-fold more computationally demanding than its execution. However, in many real engineering systems on-board compute is severely limited, e.g., in the proximity of a gas turbine engine which produces large raw data streams ($>$1GB/hour) from an array of sensors. Not only is the on-board compute limiting for on-line learning, but there is also limited computational storage capacity, low data transmission bandwidth, and a high cost of transmission, all limiting the volume of data that may be transferred for remote updating of the DT. The challenge thus presents itself to both continuously monitor the asset for changes, and to update the model of its behaviour, which together help enable accurate maintenance decisions to be made in the presence of normal data drift.

To allow continuous system monitoring, an on-board DT is thus required; while to enable utilisation to higher-power remote computing for DT updates, the efficient collection and moderation of the data volume for remote transfer is needed. Sending the data most representative of the drift is an attractive proposition for prioritising data transfer. Such data must be analysed to avoid learning an incorrect behaviour from anomalous data, e.g., data collected from a damaged system. This presents the challenge of differentiating between fault anomalies and previously unseen nominal data. The former data type must be avoided when updating the DT, but the latter is required to accurately simulate the physical system in such conditions. Data collection is a further challenge faced when updating a DT. Figure 1 illustrates this challenge.

To address these challenges, we propose a data-driven approach to develop DTs for groups of computationally resource-limited systems. Physical systems, which exhibit non-linear dynamic behaviour, are modelled using a deep neural network trained with data from each system. To update the DT over time, segments of data that are not well understood by the digital model, e.g., data generated by the system in new, previously unseen, operating conditions, are selected and used to update the model. This approach avoids the costly transmission of data with low additional information content relative to that already encapsulated within the DT model. Anomalous segments of data are automatically labelled using the fleet data set and made available for expert evaluation to aid root-cause analysis; crucially, these segments are not used to update the DT unless relabelled as normal by the expert. In contrast to model-based methods, which require physical knowledge of the system, data-driven DTs only rely on data to accurately represent their physical twins. Therefore, our framework can be used to represent a variety of complex systems.

1.1. Data Selection

The DT should be routinely updated with data generated by the physical system in order to keep the twin up-to-date. For resource-limited systems, such an update cannot be performed online; the data must be stored or transmitted to update the DT on a more powerful system. Several factors can limit the amount of data collected by resource-limited systems, e.g., data transmission cost, limited bandwidth or limited storage. All are common problems found in several applications such as wireless sensing networks (Harb & Makhoul, 2017) and surveillance systems (Begishev et al., 2018). This problem requires the development of methods to select high-quality data that are information-rich for the desired task. Solutions to the problem of selecting data include adaptive sampling rate/transmission rate (Habib, Makhoul, Darazi, & Salim, 2016), data redundancy detection (Kong et al., 2016), etc. Other methods use ideas of active learning to select the most useful data for a particular task, to be transmitted by remote sensors (Xu & Zheng, 2017).

Active learning considers the problem of selecting the fewest unlabelled data points to be labelled for the purpose of model training. The main challenge is how to select the most informative data to label such that the performance of a model is maximised. This task is especially important for deep learning due to the larger amount of data required for training. Active learning methods can be divided into three different categories (Ren et al., 2021): query synthesis, stream-based selective sampling, and pool-based. In query synthesis, it is assumed that any data point in the input space can be queried, i.e., request a label, including samples generated by the learner. Stream-based selective sampling is mostly applied to applications with resource-constrained devices, i.e., limited storage and computational power. Pool-based methods assume the opposite: that large devices with enough computing and storage resources are available. Here, at each iteration, samples from the pool are selected to be labelled based on the knowledge previously acquired.

In all of the methods, the design of the data selection or query rules is crucial. A common criterion to select data points is uncertainty (He, Jin, Ding, Yi, & Yan, 2019; Ranganathan, Venkateswara, Chakraborty, & Panchanathan, 2017; Ducoffe & Precioso, 2018). Samples whose class assignment or prediction are the most uncertain are selected for labelling. A problem with the uncertainty approach in deep learning is the estimation of uncertainty. This has been mainly addressed by using Bayesian deep learning (Gal, Islam, & Ghahramani, 2017; Pop & Fulop, 2018; Kirsch, Van Amersfoort, & Gal, 2019). Another limitation of this approach is that selecting samples based on uncertainty can result in sampling bias (Dasgupta, 2011). Most of these methods focus on the classification problem and use elements exclusive to this task such as the decision boundary.

Another criterion to select samples is diversity. In this case, the similarity between points is measured (Brinker, 2003; Yin et al., 2017). This has also been applied to the regression problem (Wu, Lin, & Huang, 2019). Selection based on diversity can lead to querying samples with low information content if not done carefully (Ren et al., 2021). To mitigate this problem, hybrid methods that combine uncertainty and diversity have been proposed (Wang, Du, Zhang, & Zhang, 2016). Density-based methods select points to attempt to find a subset representing the distribution of the entire dataset. In other words, the method tries to make the labelled and unlabelled datasets indistinguishable in terms of distribution (Gissin & Shalev-Shwartz, 2019). Finally, the expected change in the model can also be used to select samples. This change can be considered in the output (Freytag, Rodner, & Denzler, 2014), gradient change of the loss function (Cai, Zhang, & Zhou, 2013), etc. Alternatively, a network can be trained to predict the loss of an input point (Yoo & Kweon, 2019). Once trained, this network can be used to select data points.

1.2. Model Update

When the physical system changes its behaviour due to factors such as degradation, the DT must be updated to describe the new behaviour. Depending on the task, a different model update strategy is necessary. For instance, a time series forecasting model requires recent data and regular updates to capture seasonal patterns, whereas an image classification model may only need infrequent updates, focusing more on coverage across classes. Here, we are interested in prediction with time series data. Depending on the update strategy, the deep neural network has to be updated with a relatively small dataset compared to the dataset initially used to train the model. This problem has been addressed in transfer learning by using fine-tuning (Yosinski, Clune, Bengio, & Lipson, 2014). Two major problems of using a small dataset to update a pre-trained model are overfitting and catastrophic forgetting (McCloskey & Cohen, 1989), i.e., a model can forget previously learnt knowledge when it is trained with new information.

Updating strategies can be broadly categorised into two main approaches: retraining, which may be conducted either partially or fully from scratch, and parameter recalibration, where model parameters are updated incrementally without full retraining. Some methods combine elements of both approaches and are considered as hybrid strategies. In retraining approaches, models are retrained at a fixed interval by using historical data windows based on performance (Gemaque, Costa, Giusti, & Dos Santos, 2020; Richard & Belacel, 2024). This approach is suitable for problems with predictable data shifts such as in financial forecasting. The difficulty in this ap-

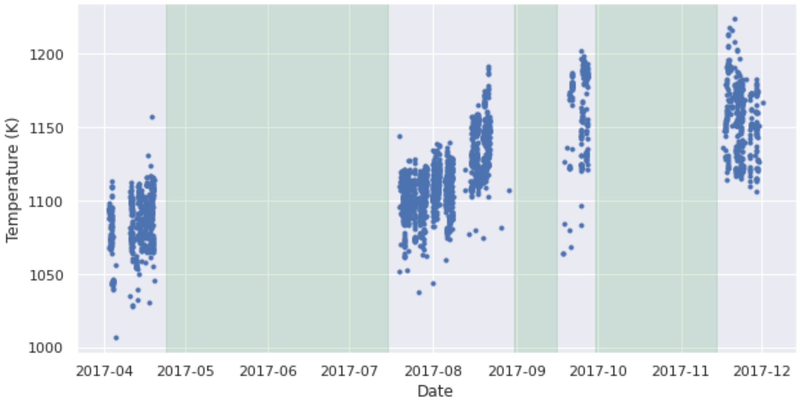

Figure 2. Mean engine temperature recorded at a fixed shaft speed. As the engine degrades over time it becomes less efficient and more energy is required to achieve a demanded shaft speed. As a result, for a given shaft speed, a higher engine temperature is recorded. Highlighted regions show periods where a major maintenance action was performed.

proach is the selection of a data window that represents the drift. In parameter recalibration methods, model parameters are updated progressively on new data over time (Lu et al., 2018; Lyu, Li, Jiang, & Hassan, 2024). This method is useful for tasks where the model has to be updated frequently or in real time. In both approaches, the main challenge is to update the model without forgetting previously learnt information (Van de Ven, Soures, & Kudithipudi, 2024). Hybrid methods combine predictions from multiple models trained on batches from different time periods. This category includes transfer learning where a pre-trained model is used and tuned to reflect the new system behaviour.

The rest of the paper is organised as follows. Section 2 presents the problem solved in this paper and introduces a case study. Section 3 explains in detail the proposed solution. Finally, results and conclusions are given in sections 4 and 5, respectively.

2. PROBLEM DEFINITION

In this paper, we focus on the development of a data-driven DT for groups of resource-limited systems. Specifically, we consider systems with the following characteristics. 1) The system is not capable of executing tasks with a high computational load such as running full-physics models, updating digital models or executing complex anomaly detection methods. 2) The system cannot store or transmit all the data to be processed by a system with more resource availability. 3) Data cannot be continuously transmitted to another system.

We consider complex non-linear and dynamic systems subject to slow nominal intrinsic degradation, rare but acceptable extrinsic disturbances and the risk of abnormal intrinsic anomalies / faults. Given these characteristics, the challenge is to develop a framework to compute and update a DT that is capable of accurately simulating a time-varying physical system throughout its lifetime. To update a DT, data must be collected from the physical system. To avoid learning incorrect behaviours from anomalous data, the data must be analysed before updating the DT. This is a challenge faced in online learning where a model is constantly updated from a data stream. The model must be updated to learn changes in data distribution resulting from factors such as system degradation, while learning anomalous behaviours must be avoided. Solutions have been proposed to distinguish between anomalies and changes in the data distribution in an online fashion (Guo et al., 2016; Saurav et al., 2018). In these approaches, when a model prediction deviates from the observed data over a significant period of time, it is assumed that the data distribution has changed. However, depending on the problem, a permanent change of the system behaviour can be considered as an anomaly. In general, a model prediction can deviate from the observed data due to the following reasons: (1) degradation: the system behaviour changes due to expected degradation, (2) unseen operating conditions: the system operates under conditions not previously observed and (3) system fault: unexpected behaviour is observed caused by a system fault. To maintain a digital representation of the system, the problems of selecting, managing, and analysing collected data must be addressed.

2.1. Case Study

The solution presented in this paper is demonstrated with a real data case study from aerospace gas turbine engines. This case study demonstrates the ability of the proposed solution to update the DT to accurately simulate a physical system when its behaviour changes through time. Data were recorded from an aerospace gas turbine engine over multiple runs on a test bed over a time period of several months. The dataset consists of 1 Hz time series data with 12 channels - each channel records data from a different sensor placed on the engine.

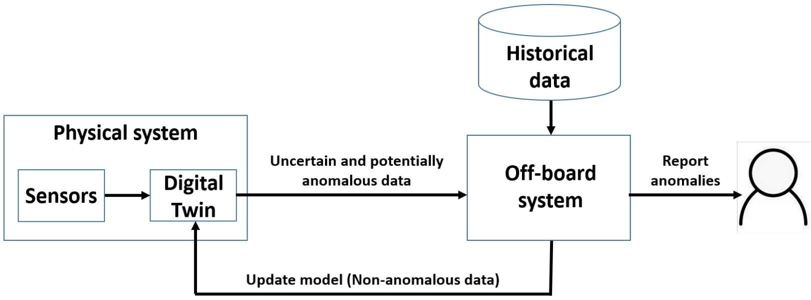

Figure 3. Overview of proposed solution. Uncertain and potentially anomalous data are selected on-board the physical system using the DT. Data are sent to the off-board system for analysis, with access to the group's historical data. The off-board system identifies and presents the anomalous data to a user for further analyses. Normal (non-anomalous) data are used to update the DT.

During this period the engine showed signs of nominal degradation and several maintenance actions were performed, see Fig. 2, where efficiency changes are seen to result in different fuel flow and temperatures being observed. The degradation and maintenance caused the dynamic behaviour of the system to change continuously over the time period.

3. SOLUTION

We propose a closed-loop approach: Data collected from the physical system are sent to an anomaly detector, which also has access to whole-group data. This allows us to identify behaviours caused by degradation or previously unseen operating conditions, for a particular system, that could be flagged as anomalous - even if such behaviour is normal at the group level. Anomalous data caused by a system fault are removed from the training data, which is then used to update the asset-specific DT, see Fig. 3. This step is necessary to avoid learning behaviour from a faulty system that is expected to be repaired.

Since we consider resource-constrained systems, collecting or transmitting all the data is not feasible nor is storing and updating group data on-asset. We use a light-weight DT capable of running on-board the physical system. The twin has the ability to identify and collect both potentially anomalous data, and data that are not well understood by the DT, i.e., it is associated with a high prediction uncertainty. An example of the latter includes data at previously unseen operating conditions. In this paper, we refer to the combination of the DT and software used for data collection as the on-board system. Data collected by the on-board system are then analysed by a centralised system, referred to as the off-board system, with access to greater computational resources and the group's historical data. The off-board system identifies whether the data returned by the on-board system are anomalous at the group level. These data can then be presented to experts for assessment of label accuracy, root-cause analysis and sanctioning of alerts. Data not identified as anomalous are used to update the DT. The two layers of anomaly detection are designed to limit the workload placed on experts to acceptable levels. In the remainder of this section, the on-board and off-board systems are presented.

3.1. On-Board Detection and Selection System

In this subsection, we present an overview of the DT model used for data selection. The on-board system consists of a lightweight deep neural network designed to run within the computational and storage constraints of the physical asset. Rather than relying on computationally intensive physics-based models, the approach uses an efficient data-driven architecture capable of real-time execution. The network processes multivariate time-series sensor data and predicts both the mean and variance of a target signal distribution. The standardised Euclidean distance between observed and predicted values is used to determine anomalous data. This probabilistic formulation enables not only accurate modelling of normal system behaviour but also an assessment of predictive uncertainty. Full architectural details and their effectiveness to detect anomalies are presented in (Hartwell et al., 2024).

Due to limited on-board storage, once the storage buffer is full, data with low novelty and uncertain are replaced as new and more informative data are received over time. A subset of the storage buffer is sized to allow its transfer over the available communication bandwidth within the operational turnaround time once the aircraft has landed. For data selection, a real-time decision is made based on the limited stored data. This means that, from the active learning perspective, our problem lies between the stream-based and pool-based categories. As presented in Section 1.1, uncertainty is a common criterion for data selection. In our problem, uncertainty reflects input conditions insufficiently represented during training, such as previously unseen operating regimes. We also consider data segments with a high anomaly score. These segments of data are post-processed as described in the next section. If a data point is later identified as non-anomalous, i.e., false positive, this is used to retrain the model. Over time, after a drift, the DT returns high scores for non-anomalous data with high confidence, i.e., different behaviour for previously seen conditions. This, in combination with the data selected based on uncertainty, increases the data diversity of the training dataset by including points far from the learnt data manifold.

3.2. Off-Board System: Data Appraisal

The primary objective of the off-board system is to identify anomalies in the data returned by the on-board system, thus maximising the use of returned data for updating the model, without learning fault behaviours. A second objective of the off-board system is to reduce the user's workload. Whilst the data volume returned by the on-board system is orders of magnitude less than all sensor data, a relatively large amount of data may be returned if an economic communication means is available (for example via 'Gatelink' technology). Successful identification of only the anomalous data significantly reduces user workload, as only a small fraction of all the data need to be reviewed. The final objective of the off-board system is to collect non-anomalous data to update the digital model, as discussed below.

Two key elements differentiate the two systems: (1) access to more computational resources and (2) access to the individual asset's and group's historical data. More computational resources mean that the off-board system can run more computationally demanding anomaly detection routines. Moreover, running time is not crucial in the off-board system in contrast to the on-board system, where data are constantly received and have to be analysed online. Access to the group's historical data allows the off-board system to identify data that are anomalous to a particular system but not to the group.

The selection of such methods is highly problem dependent. For this work, a set of algorithms was selected based on the improvement they were found to give during a training phase. Therefore, we only give an overview of the method used for the case studies where multivariate time series are analysed.

As discussed in Section 2, the data analysed in the case studies are multivariate time series. Features were extracted from the time series of all historical data. Specifically, a combination of features, obtained using Kernel PCA (Yang & Shahabi, 2005), that reflect the correlation between signals and statistical features from individual time series were used. These statistical features can be designed and updated based on knowledge from anomalous data collected by the off-board system over time. Since computational resources are not limited as in the on-board system, additional recorded signals, which may be needed to improve the model of normality of the physical system, can be considered in the off-board system. The extraction of these features is computationally expensive; hence such a method cannot be run on-board. The features are used to train a one-class Support Vector Machine (SVM). The SVM is trained in such a way that rare nominal events not learnt by the on-board system, and hence flagged as possible anomalies, are classified as non-anomalous data. When new data are received by the off-board system, features are extracted from the time series and classified by the SVM.

3.3. Off-Board System: Digital Twin Update

To maintain a good anomaly detection performance, when the physical system changes its behaviour due to factors such as degradation, see Fig. 2, the DT must be updated to describe the new behaviour. To update the model, the most recent data, representative of the current system behaviour, can be added to the previously used training dataset. While this approach avoids the forgetting of previous system behaviours, see Section 1.2, this training process can be time-consuming and computationally expensive. The other alternative is to only use the most recent data to update the model parameters. This new dataset is relatively small compared to the dataset initially used to train the model, introducing the risk of overfitting. To avoid the problem of catastrophic forgetting, we use a parameter regularisation method. By restricting the ability of the network to learn, the problem of overfitting is reduced. Formally, these methods minimise a loss function of the form:

$$\tilde{\mathcal{L}} = \mathcal{L} + \sum_{k} \Omega_k \|\theta_k - \theta_k^*\|_2^2, \tag{1}$$

where $\mathcal{L}$ is the original loss function, e.g., mean squared error, θ are the network weights, θ^* are the weights of the pre-trained network and Ω is the regularisation strength. One of the most common types of regularisation is L^2 regularisation, where all the parameters are forced towards zero, i.e., $\theta_k^* = 0 \; \forall \; k$.

Two different regularisers were considered to update the model when new data are received from the on-board system: L^2 and L^2-SP (Daumé III, 2009; X. Li, Grandvalet, & Davoine, 2018), where SP denotes starting point, referring to the pre-trained weights from which deviation is penalised. Results obtained from several tasks show that these regularisers are competitive compared to more complex and computationally demanding approaches (X. Li et al., 2018). The L^2-SP regulariser penalises deviations of all the parameters with the same factor, i.e., $\Omega_k = \alpha \; \forall \; k$ and $\alpha \in \mathbb{R}$, from the pre-trained network parameters. All the weights are hence forced to remain close to those of the pre-trained network. In contrast to a model updated with the L^2 regulariser, a model updated with the L^2-SP regulariser is expected to remember previously seen behaviours while learning new information.

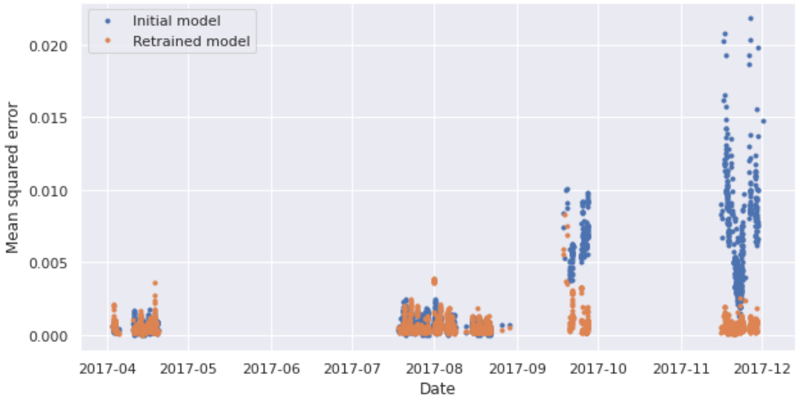

Figure 4. Average mean squared error of predictions made by the initial and iteratively retrained model across all the engine outputs.

4. RESULTS

In this section, we present the results of updating the digital model with data collected from the physical system. We also compare the regularisation approaches presented in Section 3.3 in terms of prediction accuracy and the capacity to remember previous behaviours.

An iterative update strategy is used to update the DT. An initial model is trained with all data from runs performed before April 2017, and is then used to select the unusual data from the following 10 runs (after the training data). The collected data were used to update the DT. Once the model was updated, data from the next 10 runs were processed and collected to again update the DT. This procedure was repeated until all runs were processed. The prediction errors of the initial model and the updated model are shown in Fig. 4. The significant increase in error observed after September 2017 for the initial model is due to a maintenance action on the engine. The results show that by updating the DT the behaviour of the physical system can be predicted, despite changes in the system.

To illustrate how the incorrect simulation of the physical system can affect the identification of anomalies, synthetic anomalies (spikes of fixed length and amplitude based on the signal's local standard deviation) were injected in randomly selected runs. The runs were divided into two groups: before and after the September maintenance event. As shown in Fig. 4, when the DT is not updated, the behaviour of the system is not predicted accurately. Therefore, poor performance in the detection of anomalies is expected. When the DT is not updated, 22 of the synthetic anomalies are detected before the maintenance event and only one after. In contrast, when it is updated with the L^2-SP regulariser, 28 and 23 anomalies are detected before and after the overhaul, respectively. The results show an improvement in the number of anomalies detected when the DT is routinely updated. As expected, a more noticeable improvement is made on the data after the maintenance event.

The results presented above show that when a model is routinely updated, accurate prediction of engine behaviour is achieved. However, they do not show how the model is affected in terms of forgetting previous behaviours. To see this effect, two different manoeuvres (operating profiles), labelled as A and B, were extracted from each engine run, see Fig. 5 (top). Note that only two manoeuvres were selected for simplicity. These two different manoeuvres can be seen as different engine behaviours. That is, initially, only behaviour A is observed. Over time, as the engine changes, manoeuvre B is observed instead. Note that these manoeuvres represent different behaviours for illustrative purposes and are not intended to reflect realistic changes in engine dynamics over time. The initial model was trained only with manoeuvre A. Then, the model was retrained with data from manoeuvre B following the procedure presented above. This represents the collection of new data from an engine that behaves differently from the initial training distribution. We tested three different approaches to retrain the model: the L^2 regularisation, the $L^2 - SP$ regularisation, and an augmented training data approach that includes historic data from manoeuvre A. Fig. 5 (bottom) shows the prediction of manoeuvre A after retraining the model. The prediction of manoeuvre B did not vary significantly between approaches and is not shown.

For reference, the 'initial prediction' of manoeuvre A, made by the model trained with only this manoeuvre, is included in the results. The results show that when the model is updated with the L^2 regulariser, the model forgets how to predict manoeuvre A. The forgetting is significantly reduced when using the $L^2 - SP$ regulariser. Adding data from manoeuvre A during the retraining improves the capacity of the model to remember manoeuvre A. While this approach does not repre-

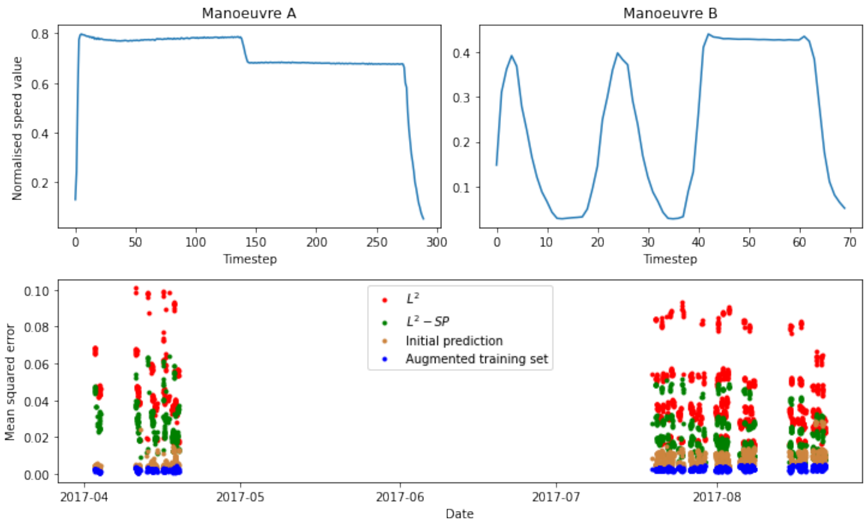

Figure 5. Top: manoeuvres representing different behaviours of the engine. Bottom: Prediction of manoeuvre A after retraining a model with manoeuvre B using different training approaches.

sent a challenge in the toy example presented above, it raises new challenges when there is a continuum of manoeuvre diversity over multiple engine signals and data must be selected from a large pool of historical engine behaviours to represent the current system behaviour.

5. CONCLUSION

Digital Twins are used in a wide range of areas for management, optimisation and decision support of physical systems. To accurately represent their physical counterparts, DTs must be routinely updated with data collected from the physical system. This can be a significant challenge for systems with constrained resources. Moreover, not all the collected data are suitable to update the DT, e.g., anomalies related to emerging faults must be handled carefully. This paper presents a framework to keep a data-driven DT up-to-date by updating it with data selected and monitored from the physical system. The proposed solution allows real-time asset monitoring and the selection of high-quality data for remote update of a twin using fault-free fleet data. Results from a gas turbine engine case show the capacity of the solution to accurately simulate the behaviour of an engine throughout its life. To further reduce the computational and data requirements for updating, interesting future research directions include the definition of (optimal) criteria on when to update the model and the introduction of physics-based constraints to the machine learning update process.

ACKNOWLEDGMENT

This work was partly funded by the Aerospace Technology Institute under the REPLENISH project.

REFERENCES

Begishev, V., Petrov, V., Samuylov, A., Moltchanov, D., Andreev, S., Koucheryavy, Y., & Samouylov, K. (2018). Resource Allocation and Sharing for Heterogeneous Data Collection over Conventional 3GPP LTE and Emerging NB-IoT Technologies. *Computer Communications*, *120*, 93–101.

Brinker, K. (2003). Incorporating Diversity in Active Learning with Support Vector Machines. In *Proceedings of the 20th International Conference on Machine Learning* (pp. 59–66).

Cai, W., Zhang, Y., & Zhou, J. (2013). Maximizing Expected Model Change for Active Learning in Regression. In *Proc. of ICDM* (pp. 51–60).

Dasgupta, S. (2011). Two Faces of Active Learning. *Theoretical Computer Science*, *412*(19), 1767–1781.

Daumé III, H. (2009). Frustratingly Easy Domain Adaptation. *arXiv Preprint arXiv:0907.1815*.

The DTHub. (2026). https://digitaltwinhub.co.uk/. (Accessed: 2026-02-20)

Ducoffe, M., & Precioso, F. (2018). Adversarial Active Learning for Deep Networks: A Margin Based Approach. *arXiv Preprint arXiv:1802.09841*.

Freytag, A., Rodner, E., & Denzler, J. (2014). Selecting Influential Examples: Active Learning with Expected Model Output Changes. In *European Conference on Computer Vision* (pp. 562–577).

Gal, Y., Islam, R., & Ghahramani, Z. (2017). Deep Bayesian Active Learning with Image Data. In *International Conference on Machine Learning* (pp. 1183–1192).

Gemaque, R. N., Costa, A. F. J., Giusti, R., & Dos Santos, E. M. (2020). An Overview of Unsupervised Drift

Detection Methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *10*(6), e1381.

Gissin, D., & Shalev-Shwartz, S. (2019). Discriminative Active Learning. *arXiv Preprint arXiv:1907.06347*.

Glaessgen, E., & Stargel, D. (2012). The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. In *Proc. of AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* (p. 118).

Guo, T., Xu, Z., Yao, X., Chen, H., Aberer, K., & Funaya, K. (2016). Robust Online Time Series Prediction with Recurrent Neural Networks. In *Proc. of DSAA* (pp. 816–825).

Habib, C., Makhoul, A., Darazi, R., & Salim, C. (2016). Self-adaptive Data Collection and Fusion for Health Monitoring Based on Body Sensor Networks. *IEEE Transactions on Industrial Informatics*, *12*(6), 2342–2352.

Harb, H., & Makhoul, A. (2017). Energy-efficient Sensor Data Collection Approach for Industrial Process Monitoring. *IEEE Transactions on Industrial Informatics*, *14*(2), 661–672.

Hartwell, A., Montana, F., Jacobs, W., Kadirkamanathan, V., Ameri, N., & Mills, A. (2024). Distributed Digital Twins for Health Monitoring: Resource Constrained Aero-engine Fleet Management. *The Aeronautical Journal*, *128*(1325), 1556–1575.

He, T., Jin, X., Ding, G., Yi, L., & Yan, C. (2019). Towards Better Uncertainty Sampling: Active Learning with Multiple Views for Deep Convolutional Neural Network. In *2019 IEEE International Conference on Multimedia and Expo* (pp. 1360–1365).

Kirsch, A., Van Amersfoort, J., & Gal, Y. (2019). Batchbald: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. *Advances in Neural Information Processing Systems*, *32*.

Kong, L., Zhang, D., He, Z., Xiang, Q., Wan, J., & Tao, M. (2016). Embracing Big Data with Compressive Sensing: A Green Approach in Industrial Wireless Networks. *IEEE Communications Magazine*, *54*(10), 53–59.

Li, C., Mahadevan, S., Ling, Y., Choze, S., & Wang, L. (2017). Dynamic Bayesian Network for Aircraft Wing Health Monitoring Digital Twin. *AIAA Journal*, *55*(3), 930–941.

Li, X., Grandvalet, Y., & Davoine, F. (2018). Explicit Inductive Bias for Transfer Learning with Convolutional Networks. *arXiv Preprint arXiv:1802.01483*.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., & Zhang, G. (2018). Learning under Concept Drift: A Review. *IEEE Transactions on Knowledge and Data Engineering*, *31*(12), 2346–2363.

Lyu, Y., Li, H., Jiang, Z. M., & Hassan, A. E. (2024). On the Model Update Strategies for Supervised Learning in Aiops Solutions. *ACM Transactions on Software Engineering and Methodology*, *33*(7), 1–38.

Magargle, R., Johnson, L., Mandloi, P., Davoudabadi, P., Kesarkar, O., Krishnaswamy, S., … Pitchaikani, A. (2017). A Simulation-based Digital Twin for Model-driven Health Monitoring and Predictive Maintenance of an Automotive Braking System. In *Proc. of the International Modelica Conference* (pp. 35–46).

McCloskey, M., & Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation* (Vol. 24, pp. 109–165). Elsevier.

Pop, R., & Fulop, P. (2018). Deep Ensemble Bayesian Active Learning: Addressing the Mode Collapse Issue in Monte Carlo Dropout via Ensembles. *arXiv Preprint arXiv:1811.03897*.

Qi, Q., & Tao, F. (2018). Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. *IEEE Access*, *6*, 3585–3593.

Ranganathan, H., Venkateswara, H., Chakraborty, S., & Panchanathan, S. (2017). Deep Active Learning for Image Classification. In *2017 IEEE International Conference on Image Processing* (pp. 3934–3938).

Ren, P., Xiao, Y., Chang, X., Huang, P.-Y., Li, Z., Gupta, B. B., … Wang, X. (2021). A Survey of Deep Active Learning. *ACM Computing Surveys*, *54*(9), 1–40.

Richard, R., & Belacel, N. (2024). A Conditioned Unsupervised Regression Framework Attuned to the Dynamic Nature of Data Streams. In *Proceedings of the Future Technologies Conference* (pp. 1–23).

Saurav, S., Malhotra, P., TV, V., Gugulothu, N., Vig, L., Agarwal, P., & Shroff, G. (2018). Online Anomaly Detection with Concept Drift Adaptation Using Recurrent Neural Networks. In *Proc. of the ACM India Joint International Conference on Data Science and Management of Data* (pp. 78–87).

Tao, F., Zhang, H., Liu, A., & Nee, A. Y. (2018). Digital Twin in Industry: State-of-the-art. *IEEE Transactions on Industrial Informatics*, *15*(4), 2405–2415.

Tuegel, E. J., Ingraffea, A. R., Eason, T. G., & Spottswood, S. M. (2011). Reengineering Aircraft Structural Life Prediction Using a Digital Twin. *International Journal of Aerospace Engineering*, *2011*.

Van de Ven, G. M., Soures, N., & Kudithipudi, D. (2024). Continual Learning and Catastrophic Forgetting. *arXiv Preprint arXiv:2403.05175*.

Wang, Z., Du, B., Zhang, L., & Zhang, L. (2016). A Batch-mode Active Learning Framework by Querying Discriminative and Representative Samples for Hyperspectral Image Classification. *Neurocomputing*, *179*, 88–100.

Wu, D., Lin, C.-T., & Huang, J. (2019). Active Learning for Regression Using Greedy Sampling. *Information Sciences*, *474*, 90–105.

Xu, Q., & Zheng, R. (2017). When Data Acquisition Meets

Data Analytics: A Distributed Active Learning Framework for Optimal Budgeted Mobile Crowdsensing. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications* (pp. 1–9).

Yang, K., & Shahabi, C. (2005). A Pca-based Kernel for Kernel Pca on Multivariate Time Series. In *Proc. of ICDM*.

Yin, C., Qian, B., Cao, S., Li, X., Wei, J., Zheng, Q., & Davidson, I. (2017). Deep Similarity-based Batch Mode Active Learning with Exploration-exploitation. In *2017 IEEE International Conference on Data Mining* (pp. 575–584).

Yoo, D., & Kweon, I. S. (2019). Learning Loss for Active Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 93–102).

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How Transferable Are Features in Deep Neural Networks? In *Advances In Neural Information Processing Systems* (pp. 3320–3328).