Using Explainable Artificial Intelligence to Interpret Remaining Useful Life Estimation with Gated Recurrent Unit

Marcia L. Baptista[1], Madhav Mishra[2], Elsa Henriques[3], and Helmut Prendinger[4]

[1] *NOVA Information Management School (NOVA IMS),*
Universidade Nova de Lisboa, Campus de Campolide,
1070-312 Lisboa, Portugal
m.baptista@novaims.unl.pt

[2] *RISE Research Institutes of Sweden, Mölndal,SE-431 53, Sweden*
madhav.mishra@ri.se

[3] *University of Lisbon, Instituto Superior Tecnico,Lisbon, Portugal*
elsa.h@tecnico.ulisboa.pt

[4] *National Institute of Informatics,Tokyo, Japan*
helmut@nii.ac.jp

Abstract

In engineering, prognostics can be defined as the estimation of the remaining useful life of a system given current and past health conditions. This field has drawn attention from research, industry, and government as this kind of technology can help improve efficiency and lower the costs of maintenance in a variety of technical applications. An approach to prognostics that has gained increasing attention is the use of data-driven methods. These methods typically use pattern recognition and machine learning to estimate the residual life of equipment based on historical data. Despite their promising results, a major disadvantage is that it is difficult to interpret this kind of methodologies, that is, to understand why a certain prediction of remaining useful life was made at a certain point in time. Nevertheless, the interpretability of these models could facilitate the use of data-driven prognostics in different domains such as aeronautics, manufacturing, and energy, areas where certification is critical. To help address this issue, we use Local Interpretable Model-agnostic Explanations (LIME) from the field of eXplainable Artificial Intelligence (XAI) to analyze the prognostics of a Gated Recurrent Unit (GRU) on the C-MAPSS data. We select the GRU as this is a deep learning model that a) has an explicit temporal dimension and b) has shown promising results in the field of prognostics and c) is of simplified nature compared to other recurrent networks. Our results suggest that it is possible to infer the feature importance for the GRU both globally (for the entire model) and locally (for a given RUL prediction) with LIME.

1. Introduction

Improved reliability is one of the key drivers of the development of more efficient maintenance strategies (J. Lee, Holgado, Kao, & Macchi, 2014). The vision here is to have machines that can monitor themselves and alert the operator ahead of time of future maintenance needs to maximize function time and avert failure. The framework behind this vision is that of Reliability Centered Maintenance (RCM) (NASA, RCM, 2008), a discipline that aims to propose tools and practices to better monitor, predict and understand the behavior of physical assets (Moubray, 2001). Major goals are to improve safety, availability, reduce logistics and maintenance costs, and to drive customer satisfaction and loyalty. Importantly, successful adoption of RCM aims to provide a greater *understanding* of the nature of the risk that is being managed.

For a given physical asset, the outcome of an RCM program is the implementation of an appropriate maintenance strategy (NASA, RCM, 2008, pp. 3-1). A strategy that many industries have followed for years due to its simplicity and generality is the preventive or time-based maintenance. In Time-Based Maintenance (TBM), repair and replacement are based on simple measures of the expected life of the equipment, such as calendar or usage time (NASA, RCM, 2008, pp.

Marcia L. Baptista et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

5-4). However, with advances in sensor technologies, preventive maintenance has been giving place to Predictive Maintenance (PM). This advanced form of maintenance uses mainly non-intrusive and non-destructive monitoring technologies to assess performance and detect defects before actual failure (NASA, RCM, 2008, pp. 5-7). Predictive algorithms are used in the process of establishing the real condition of a machine or equipment. Hence, the designation of *predictive* maintenance.

One of the most challenging and beneficial aspects of RCM and PM is prognostics. The engineering discipline of prognostics concerns the forecasting of an event of interest based on the current and past condition data (Bonissone & Goebel, 1999). For example, it can be the estimation of the Remaining Useful Life (RUL), failure margin or overall performance prediction. The maturity of condition monitoring technology coupled with the significance of prognostics has led to an increasing interest in the field over the past few years (Nguyen et al., 2019; Ucar, Karakose, & Kırımça, 2024).

Prognostics can be performed in two approaches, model-based and data-driven (Jardine, Lin, & Banjevic, 2006). Model-based methods exploit domain knowledge of the system and its failure mechanisms as they rely on principles from physics to describe the behavior of the assets. Despite the performance of these methods (Kulkarni, Daigle, Gorospe, & Goebel, 2018), they require extensive experimentation and verification during development. As an alternative, there are data-driven methods, which use large amounts of data to train machine learning algorithms to capture degradation trends in a *black-box* manner (Schwabacher & Goebel, 2007).

Prognostics models have the primary purpose of precisely predicting the behavior of a system. However, the quality of *interpretability* (or understandability) should also be a property of these models (Baptista, Goebel, & Henriques, 2022). In contexts such as aeronautics, energy, or other domains where safety plays a critical role, it is often necessary to provide an explanation for the predictions of the model, and interpretability here is of relevance. Furthermore, this property is important for better understanding the underlying mechanisms of the models as well as the associated limitations and potential pitfalls (Antamis et al., 2024).

The model-based approach is often preferred in prognostics over the data-driven approach (Daigle, 2014) due to the fact that these techniques are easier to interpret due to the existence of an underlying physics model of the system (Celaya, Saxena, & Goebel, 2012), where model variables have physical meaning and promote a better understanding of the behavior of the system. In contrast, data-driven methods can be seen as *black-boxes*, because of the lack of explicit equations. Leaving not much space for interpretation, black-box systems map the input features to a target output without exposing the reasons why (Tzeng & Ma, 2005). There are a few exceptions to this rule, such as decision trees (Quinlan, 1986; Hu, Rudin, & Seltzer, 2019), but even in these techniques, which provide some disclosure of how decisions are made, it is not always clear how individual predictions are made.

To address the general lack of transparency of machine learning techniques, a number of interpretability methods have been proposed (Schoenborn & Althoff, 2019). Here, post-hoc (agnostic) methods have become increasingly popular, as they allow to explain any type of model. A post-hoc model in Explainable AI (XAI) refers to a method applied after a machine learning model is trained to interpret or explain its predictions, without altering the original model. These models provide insights into the decision-making process of complex, often black-box models by using techniques like feature importance, local explanations, or visualization tools. Post-hoc models can help explain a model around a specific input sample (locally) or for the entire model space (globally).

Local Interpretable Model-agnostic Explanations (LIME) was one of the first local agnostic models (Ribeiro, Singh, & Guestrin, 2016) from eXplainable Artificial Intelligence (XAI). The approach builds explanation models on top of a given prediction algorithm to give reasons for the decisions of individual observations. It is a local approach as it tries to isolate separately the most important factors that influence each single decision. Precisely, LIME performs small changes to a given model input, in order to isolate the most important factors that influence local decisions. It works by fitting these separate explanations (e.g., by linear regression or decision tree models) to the local neighborhood of an individual prediction.

In the field of prognostics, the topic of data-driven interpretability is not fully explored. As a contribution, our aim is to investigate the capability of LIME to interpret a deep learning prognostics model built on simulated data from the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). The C-MAPSS (Litt, Frederick, & DeCastro, 2008) is a software from the National Aerospace Space Agency (NASA) to model the operation of a commercial turbofan engine. Data are provided by the Center for Excellence in Prognostics (CoE) of NASA Research Ames (Saxena, Goebel, Simon, & Eklund, 2008).

As the prognostics modeling approach for this study, we have selected the Gated Recurrent Unit (GRU) (Cho et al., 2014). This model has the advantage of dealing with the temporal dimension of sensor data explicitly. There are other deep architectures for sequence modeling, such as Long-Short Term Memory Networks (LSTM) (Hochreiter & Schmidhuber, 1997) or Echo State Network (ESN) (Jaeger, 2001, 2002). However, GRU is a simple and efficient alternative that has already shown promising prognostic results. For example, Hasib, Rahman, Khabir, and Shawon (2024) studied three types of recurrent networks (GRU, Bi-LSTM, and LSTM) and reported that the GRU was the best performing model on C-MAPSS.

The remainder of this article is organized as follows. Section 2 describes related work. We review work in the field of eXplainable Artificial Intelligence (XAI) and include a review on the use of the GRU in prognostics. Section 3 presents the modeling framework. We describe briefly the underlying principles of LIME and how we applied it to our GRU model. Results of the experiments are presented and discussed in Section 4 while Section 5 concludes the article.

2. RELATED WORK

Model *interpretability* (comprehensibility or understandability) is an important aspect to prognostics, where decision-making is often dependent on the understanding of forecasts. The *black-box* nature of the machine learning used in the data-driven approach has often led to mistrust and a lack of understanding of this type of methods (Erasmus, Brunet, & Fisher, 2021). Exposing explanations for the predictions could help increase trust in the model and help achieve properties such as reliability and certification (Doshi-Velez & Kim, 2017) as well as understand the potential errors and limitations (Fen et al., 2019).

Model interpretability has multiple definitions (Lipton, 2018). It can be defined in general as the means to build trust (Kim, 2015). Other definitions can take on a more formal character. For example, for Lou, Caruana, and Gehrke (2012), interpretability is defined as the ability to understand the contribution of different predictors to the model. The authors use generalized additive models to capture the causal relationships between (individual) features and output. They argue that additive models are more accurate than generalized linear models and still retain the intelligibility of linear approaches. For Ribeiro et al. (2016), interpretability is the ability of a model to provide qualitative understanding between the input and the response according to the user's limitations. For example, a relatively simple decision tree may be considered interpretable, but a more complex decision tree may not be comprehensible to a human.

Lakkaraju, Bach, and Leskovec (2016) provide a more focused definition of model interpretability that applies to classification approaches. Here, the concept is defined as the ability to provide decision boundaries between classes and explain why a label is predicted in a certain way for a data point. The authors propose interpretable decision sets arguing that this approach is both accurate and interpretable. This kind of predictive models relies on sets of independent causal rules to reach a predictive decision. In this paper, we do not formalize the notion of model interpretability but instead adopt a general definition of **building trust** and **providing intuitive explanations**.

2.1. eXplainable Artificial Intelligence (XAI)

The discipline of eXplainable Artificial Intelligence (XAI) (Gunning & Aha, 2019), which studies the development of interpretable methods, has two main approaches – the design of interpretable models (Agarwal et al., 2021) or the implementation of post-hoc (agnostic) models (Turbé, Bjelogrlic, Lovis, & Mengaldo, 2023). Interpretable models are inherently transparent and understandable by design, while post-hoc methods (agnostic to the underlying predictive model) are applied after model training to explain or interpret the predictions of more complex, opaque models.

In this paper, we study Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016), a post-hoc local model. LIME creates approximate models on top of a more complex machine learning model to expose causal reasons for the prediction of individual instances. The overall goal of LIME is to select an explanation function over the interpretable space that is locally faithful to the observed model.

The potential limitations of LIME have been shown by Melis and Jaakkola (2018), whose results suggest that LIME explanations are not always stable when applied to nonlinear models. The source of LIME instability in such cases was further studied by Fen et al. (2019). The authors demonstrated the presence of three sources of uncertainty, namely randomness in sampling, variation with sampling proximity, and variation in model credibility across different data samples. Despite its limitations, LIME continues to be a significant and popular model in XAI. Since it is model-agnostic and is based on a simple and understandable idea, we opted for this approach in this paper.

Reviewing work in post-hoc interpretability models for prognostics and health management, an early work of note is that of Zeldam (2018) who applied XAI methods to fault diagnosis. The research addressed the problem of incomplete or inaccurate maintenance reports, filled with free-form text. The author proposed a custom XAI methodology to explain why a particular failure diagnosis was made by comparing the features of the failure to expected values across different fault modes.

In terms of work with post hoc XAI models, an important contribution is by Serradilla et al. (2020). This work focused on interpreting Remaining Useful Life (RUL) estimations in industrial settings with two Explainable Artificial Intelligence (XAI) techniques. Their model used Random Forest as the core machine learning model. To enhance interpretation, they applied LIME and ELI5 XAI techniques.

Kundu and Hoque (2023) highlight the limitations of post-hoc XAI methods like LIME and SHAP due to inconsistencies in feature ranking. They argue that no single explanation method is universally best for all scenarios in predictive maintenance. Instead, they propose using a trust score to quantify the reliability of each explanation method.

More recently, Gawde et al. (2024) studied the effectiveness of predictive maintenance in industrial settings, specifically for steam generators. The researchers addressed the problem of using multiple data sources to perform fault diagnostics. The key aspect of the study is the use of Explainable AI (XAI) techniques, including LIME, SHAP, PDP, and ICE. This study is different from ours in that it focused on diagnostics and not prognostics. Their study was also focused on multi-modal learning and XAI.

Another work in diagnostics that was applied to the turbofan engine dataset C-MAPSS was by Ji, Zhang, and Yan (2024). The approach aimed to predict equipment failures using interpretable AI methods such as SHAP and LIME. The main contribution was the exploration of knowledge graphs to obtain more comprehensive insights into faulty components and enhance the interpretability of machine learning in predictive maintenance applications.

A similar work to ours is by Dogga, Sathyan, and Cohen (2024). The authors studied the application of SHAP and LIME to turbofan engines on C-MAPSS. In contrast to our approach, the authors did not focus on recurrent neural networks. Balasubramani, Shi, and DeLaurentis (2024) studied in more detail the application of SHAP to turboengines using convolutional neural networks.

Hasib et al. (2024) studied the interpretability of three types of recurrent networks (GRU, Bi-LSTM, and LSTM) using LIME. This is the closest to our work, but differently to our contribution, the authors do not focus on global and local interpretability at the same time.

2.2. Recurrent Neural Networks in Prognostics

In the domain of prognostics, a data-driven method that has shown promising results when handling the temporal dimension of sensor data is Recurrent Neural Networks (RNN) (An, Kim, & Choi, 2015). This method involves some complexity and could benefit from further interpretation. Since we will apply LIME to RNNs and to better contextualize the reader, we hereafter review some of the results of RNNs in prognostics.

Several works based in Recurrent Neural Networks (RNNs) have been proposed in prognostics over the years. One of the earliest contributions (Tse & Atherton, 1999) compared a typical feedforward neural network, to a classical autoregressive model and an RNN. The three approaches were applied to the prediction of nonlinear sunspot activities and vibration fault trends of industrial machines. In both cases, the recurrent network showed superior performance. Interestingly, these results were obtained with a considerably simple recurrent neural network having four input nodes, four hidden nodes, one output node and a feedback loop linked to an extra input node.

A work of note in prognostics is that of Heimes (2008) who proposed an ensemble model of RNNs for the IEEE 2008 Prognostics and Health Management challenge. The proposed ensemble model was compared against a multi-layer perceptron network. Both models were trained using the Extended Kalman Filter method. Comparing results, the RNN models showed superior performance near the end of useful life of the equipment.

Several works have used Long-Short Term Memory (LSTMs) to estimate the RUL of aero engines (Yuan, Wu, & Lin, 2016; Dong, Li, & Sun, 2017; W. Zhang, Jin, Zhang, Zhao, & Hou, 2019) and other equipment such as lithium-ion batteries (Y. Zhang, Xiong, He, & Liu, 2017; Hinchi & Tkiouat, 2018; Long, Li, Gao, & Liu, 2019), fuel cells (Ma et al., 2018) and bearings (K. Lee, Kim, Kim, Hur, & Kim, 2018b). LSTM is a type of RNN that uses memory cells instead of recurrent units. In LSTMs each cell is updated according to the activation of gates which control the operation performed on the memory cell: write (input gate), read (output gate) and reset (forget gate).

The Gated Recurrent Unit (GRU) is another RNN that also uses the gating mechanism. It uses an update and a reset gate. The update gate determines how much the inputs can change the new state while the reset gate determines to what extent memory persists. GRUs have similar performance to LSTMs (Chung, Gulcehre, Cho, & Bengio, 2014) but have fewer parameters and a faster learning process. GRUs have been applied with success in prognostics (Song, Li, Peng, & Liu, 2018; K. Lee, Kim, Kim, Hur, & Kim, 2018a).

We choose to investigate the capability of LIME to interpret an RNN as this the RNN is one of the most complex forms of deep learning. Due to the explicit modeling of time inside the network, there are several feedback looks which make it difficult to understand why and how the approach works. Here, we hope to advance knowledge on how to use LIME in prognostics and also to improve our understanding of the specific type of RNN, the GRU.

3. MODEL

This section overviews the theory of GRU and LIME. It also describes the case study.

3.1. Gated Recurrent Unit (GRU)

We selected the Gated Recurrent Unit (GRU) as our black-box for two main reasons: 1) its capacity to deal with the temporal dimension of sensor data and 2) the popularity of Recurrent Neural Networks (RNNs) in prognostics and 3) the simplified nature of GRU compared to other recurrent networks. GRU (Gated Recurrent Unit) is also often considered better than traditional RNNs because it mitigates the vanish-

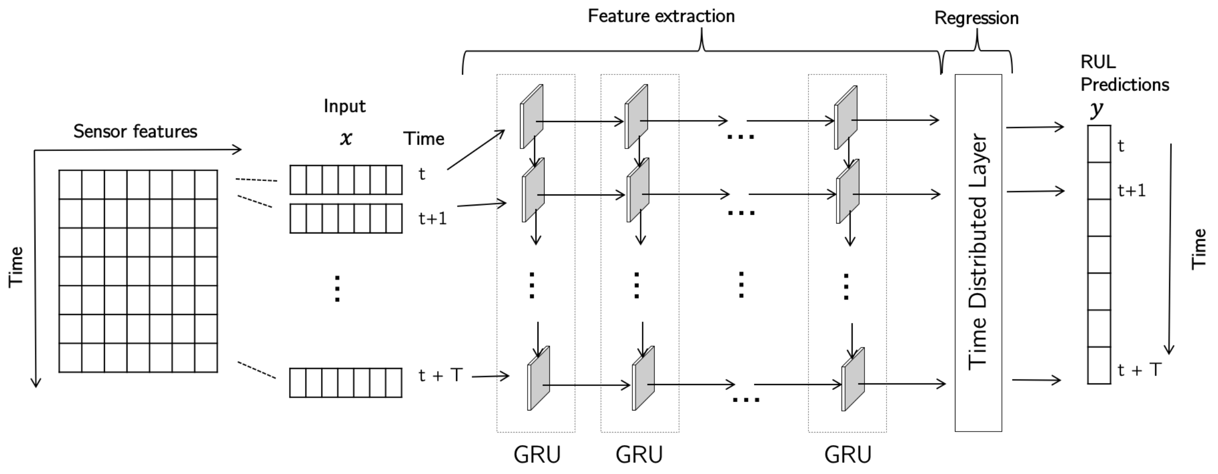


Figure 1. Gated Recurrent Unit (GRU) architecture.

ing gradient problem through gating mechanisms that control the flow of information, enabling better retention of long-term dependencies. GRUs have shown to be more computationally efficient than LSTMs as they use fewer gates, simplifying the architecture while maintaining similar performance for many sequence-based tasks.

Fig. 1 illustrates the used GRU architecture and its building blocks. The main elements are the GRU hidden layers which are used mainly to automatically learn new features from the input data and the time distributed layer which is responsible for making the final forecast.

The GRU supports different types of input to output structures. In our case, we are interested in the many-to-many architecture, i.e. predicting sequence of vectors based on time series data. In order to implement the many-to-many paradigm we used the time distributed dense layer. This layer applies a dense (fully full connected) function across every output over time to ensure that there is the same number of outputs as inputs. This is important to estimate the Remaining Useful Life (RUL) at each time step. Formally, the goal is to provide at each time step t a RUL estimation (y) given a multi-dimensional sensor input (x).

We make use of two common techniques in order to train the GRU: dropout and early stopping. We apply dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) to avoid overfitting and make the network generalize better. This means that, during training, each GRU cell can be discarded from the network according to a certain probability. Second, and also to avoid model overfitting, the networks are trained using the classical "early stopping" mechanism (Morgan & Bourlard, 1990). The data are split into training and validation sets in proportion 3:1. The validation set is used to evaluate the generalization error. Training stops when the error on the validation set is below a minimum delta. The best network is selected according to training loss.

The GRU architecture used in this paper was optimized using Talos (*Autonomio Talos [Computer Software]*, 2019). Talos is a tool which permits running hyperparameter optimization experiments. To evaluate model performance we used the randomized grid search optimization strategy (Bergstra & Bengio, 2012). The following hyperparameters were optimized: (a) number of hidden layers, (b) number of GRU cells per layer, (c) optimizer, (d) activation function of the hidden layers and (e) dropout (see Table 1). As number of timesteps, we chose a window of 10 time steps. Please note that we use stateful networks (Bulín, Šmídl, & Švec, 2019) and as such we do not need a wide time window. Within each batch of data, the network state (or memory) is maintained and it is possible to learn the relationships among the different sequences. In other words, network memory persists across the batch of data.

Table 1. Hyperparameter search ranges of Gated Recurrent Unit (GRU).

Hyperparameter	Search Range
Time steps	[10]
Hidden Layers	[1, 2, 3, 4]
Nodes per Layer	[5, 10, 25, 50]
Optimizer	[Adam, Nadam, RMSProp]
Activation	[Relu, Elu, Linear]
Output Activation	[Linear]
Dropout	[1, 0, 0.5]
Loss	[Mean Squared Error]

3.2. LIME

Given a model $f\colon X \to Y$ and an instance $x \in X$, the goal of local interpretability is to explain $f(x)$, the individual prediction of model f for data point x (Ribeiro et al., 2016). To achieve this goal, the Local Interpretable Model-agnostic Explanations (LIME) generates M explanation functions $g \in G$. The learned explanation function $g_L(x)$ is selected from functional space G as the function that minimizes loss, i.e. is a good local approximation of f, and has low complexity.

To build an explanation $g \in G$, LIME generates N samples around a given data point x. Every sample x' represents a perturbed version of x where perturbations are obtained according to a "perturbation distribution" based on π_x. The best explanation function $g_L(x)$ is selected from functional space G by solving

$$g_L(x) = \arg\min_{g \in G} \mathcal{L}(f, g, \pi_x) + \Omega(g) \qquad (1)$$

where $\mathcal{L}(f, g, \pi_x)$ is a measure of the fidelity of local model g, $\pi_x(x')$ is a proximity measure between a sample x' and x, and Ω is a penalty for model complexity.

This formulation can be used with different explanation functions g, fidelity functions $\mathcal{L}$, and complexity measures Ω. In this paper, we follow the original formulation of Ribeiro et al. (2016). Concretely, we consider G to be the class of linear models, such that

$$g(x') = w_g \cdot x' \qquad (2)$$

where w_g denotes the coefficients of the linear model g. We use the locally weighted square loss as $\mathcal{L}(f, g, \pi_x)$ and the complexity function $\Omega(g)$ is the number of non-zero weights of function g.

Concerning the perturbation distribution and as in the original work of Ribeiro et al. (2016), we use as proximity measure

$$\pi_x(x') = \exp(D(x, x')/\sigma^2) \qquad (3)$$

where $\pi_x(x')$ is an exponential kernel defined on distance D with kernel width σ. As D we use Euclidean distance and as σ we use $\frac{3}{4}\sqrt{p}$ (the recommended value by the authors in their implementation), where p is the number of input features.

Algorithm 1 illustrates how a single explanation function g is created from x. LIME builds the local model by sampling N x' instances around x. Every new instance x' is a perturbed version of x. LIME fits a weighted linear regression (WLR) around the set of points $f(x')$ weighted according to the similarity kernel function $\pi_x(x')$. In Fig. 2 we provide an overview on how LIME creates an explanation function g.

In our implementation, we consider the GRU to be the model to be interpreted (f). This model receives at each time a multidimensional input vector (x) and generates an output $y = f(x)$ which is the predicted RUL at that time step, as can be seen in Fig. 1. LIME generates M explanation functions $g \in G$ for a given x and solves equation 1 to obtain the final explanation g_L. This can be done several times for different x, each time creating a new LIME explanation. The learned function g_L intends to be a good approximation of f in the local space around x, but not necessarily a good global approximation. This is designed as local interpretability (Ribeiro et al., 2016). In this study we also investigate how to explore global interpretability using LIME.

Algorithm 1 Generation of a LIME Explanation

```
procedure LIME($f, x, N$)                      ▷ Explain sample x
    $Y' \leftarrow \{\}$
    for $i \in 1, 2, ..., N$ do
        $x'_i \leftarrow$ samplearound($x$)     ▷ Perturb features of x
        $Y' \leftarrow Y' \cup < x'_i, f(x'_i), \pi_x(x'_i) >$
    $w_g \leftarrow$ WLR($Y'$)                  ▷ Weighted Linear Regression
return $w_g$
```

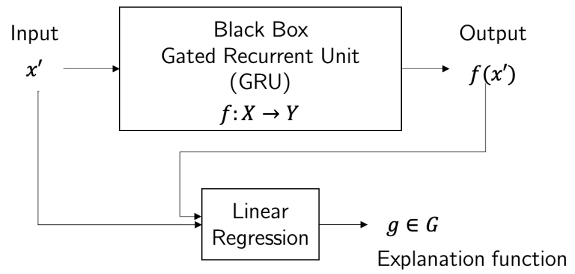


Figure 2. LIME creates an explanation function g by observing the relationship between perturbed input and output and by applying a weighted linear regression fit on $f(x')$.

3.3. Case Study

In this paper, we study jet engine data from the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS). The C-MAPSS (Litt et al., 2008) was developed by National Aerospace Space Agency (NASA) to simulate the operation of a commercial turbofan engine. It simulates an engine of the $90,000$ lb thrust class and it allows simulating (i) altitudes ranging from sea level to 40,000 ft, (ii) Mach numbers from 0 to 0.90, and (iii) sea-level temperatures from -60 to 103°F. Process and measurement noises have been added to the data resulting in complex noise dynamics. At the start of operation, each engine has a given level of wear and manufacturing variation. This wear and variation should not be considered fault condition.

In C-MAPSS data, an engine is characterized by a set of 21 prognostics sensors and 3 additional sensors (Altitude, Mach Number and Throttle Resolver Angle) indicators of operating conditions. More details about C-MAPSS data can be found in Saxena, Goebel, et al. (2008). In this work, we use the first dataset (FD001) to limit the scope of the study. In dataset FD001 there is only one operating condition which simplifies

Table 2. Prognostics features available in C-MAPSS data. The features used in this paper are highlighted in gray.

Parameter	Description	Units
T2	Total temperature at fan inlet	°R[1]
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P50/P2)	–
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	–
farB	Burner fuel-air ratio	–
htBleed	Bleed Enthalpy	–
Nfdmd	Demanded fan speed	rpm
PCNfRdmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

our analysis.

Table 2 lists all the sensor signals of C-MAPSS and highlights in gray the ones that were used to build the models. We selected 14 condition signals to monitor the effects of faults and degradation in the five rotating components of the engine: Fan, Low Pressure Compressor (LPC), High Pressure Compressor (HPC), High Pressure Turbine (HPT), and Low Pressure Turbine (LPT). Our goal here was to select a set of signals (features) which could adequately represent the degradation process.

3.4. Evaluation Measures

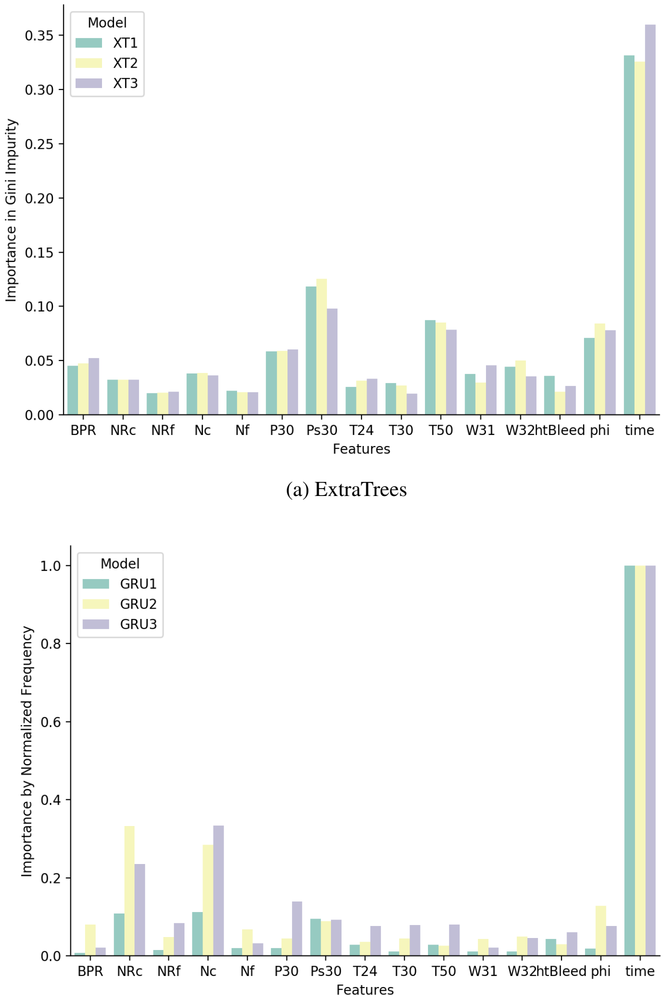
Different evaluation metrics provide distinct perspectives on how well a model fits the data. Used metrics include Root Mean Squared Error (RMSE), which measures the magnitude of prediction errors, and R-squared (R²), which assesses the proportion of variance explained by the model. Table 3 summarizes information about these two measures.

4. RESULTS

In this section, we present and discuss results. We start by examining classical feature importance ranking using Extremely Random Trees (Extratrees). The interpretability results, in local and global terms, produced by LIME are then analyzed.

Metric	Formula	Interpretation
RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	Lower values indicate fewer large errors and better performance.
R²	$1 - \frac{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}{\sum_{i=1}^{n}(y_i-\bar{y})^2}$	Higher values indicate better fit, with 1 being perfect and 0 no explanatory power.

Table 3. Evaluation Measures



(a) ExtraTrees

(b) LIME

Figure 3. Feature importance.

4.1. Baseline: Extremely Random Trees (XT)

As a baseline for LIME, we use Extremely Random Trees (ExtraTrees) (XT) (Geurts, Ernst, & Wehenkel, 2006) to globally rank the features. The ExtraTrees method is a pure randomization method that has a node splitting strategy different from that of traditional random forests (Breiman, 2001). The

[1] °R is a temperature measurement used in aeronautics.

method has been shown to be competitive on ranking problems while requiring less parameter tuning than other boosting techniques (Geurts & Louppe, 2011).

We have analyzed the ranking output of three randomly choosen ExtraTrees models (XT1, XT2 and XT3) and the results were congruent. Our intention was to analyze three distinct models and see if they reported similar results. The small variation of results from model to model gave credibility to the analysis. Please note that several other experiments were carried out to verify the stability of the results. As shown in Fig. 3a, the most important features, according to (all) the tested models, were the variables of *time*, *Ps30*, *T50*, *phi*, and *P30*. The most determinant variable, time, was expected to be this relevant, since aging is a fundamental determinant of degradation.

4.2. Global Interpretability with LIME

We were interested in studying global interpretability using LIME. To estimate feature importance and as a first approach, we calculated the number of times a feature is mentioned in a local explanation, i.e. associated to a non-zero weight, considering the first 10 weights. Three GRU models (see Table 4) were examined according to this evaluation measure. These three models were randomly chosen. Our intention was to analyze three distinct models and see if they reported similar results. Results are presented in Fig. 3b. As shown, the models agree to some extent in the importance of some features: *Nc* and *NRc* were the most frequently used by LIME to provide explanations for the GRU models. The importance of the remaining parameters was not so apparent.

To quantify how well LIME approximated the observed models and in order to measure how faithful local models are to GRU, we averaged the R^2 score of the linear regression explanations. As shown in Tab. 4, the average fidelity of LIME to the models was considerably low ($< 30\%$) and the stability of the models was also low, as measured by the standard deviation of the R^2. This result raises some concerns about the tractability of LIME.

Interestingly, the features most often used by LIME were not always the most relevant features of ExtraTrees. Some variables were important predictors both for LIME and ExtraTrees, such as the variable of time, but other features were assigned different importance. For example, *phi* is an important parameter for ExtraTrees but is almost ignored by LIME. The opposite situation also occurred. This can possibly be because the underlying observed model is different and the reasoning process may be different. This may even be true for the same architecture under different configurations and this might explain why different GRU models have different global interpretations.

Table 4. Prognostics performance measured in Root Mean Squared Error (RMSE) and LIME fidelity measured in R^2 score[2], on average, and considering a single unit.

Model	RMSE	R^2 (%)	Fidelity of Single Unit R^2 (%)		
			$t = 10$	$t = 90$	$t = 180$
GRU1	45.50	14.00±6.1	48.17	10.70	16.50
GRU2	46.65	14.12±6.2	23.98	9.14	13.92
GRU3	45.96	28.12±17.6	60.54	10.17	22.83

We also note the more "focused" reasoning of GRU. According to the interpretation in Fig. 3b, the GRU models seem to look mainly at essential features (e.g. *NRc*, *Nc* and *time*) to base their decisions. This kind of reasoning comes in line with what a human usually does and also with what a human tends to understand better. Even if this is not exactly how GRU works, the local models that LIME generates help to extract these rules that provide enhanced interpretability. Having access to these explanations, rather than the classical feature importance charts, is of relevance to the field of prognostics.

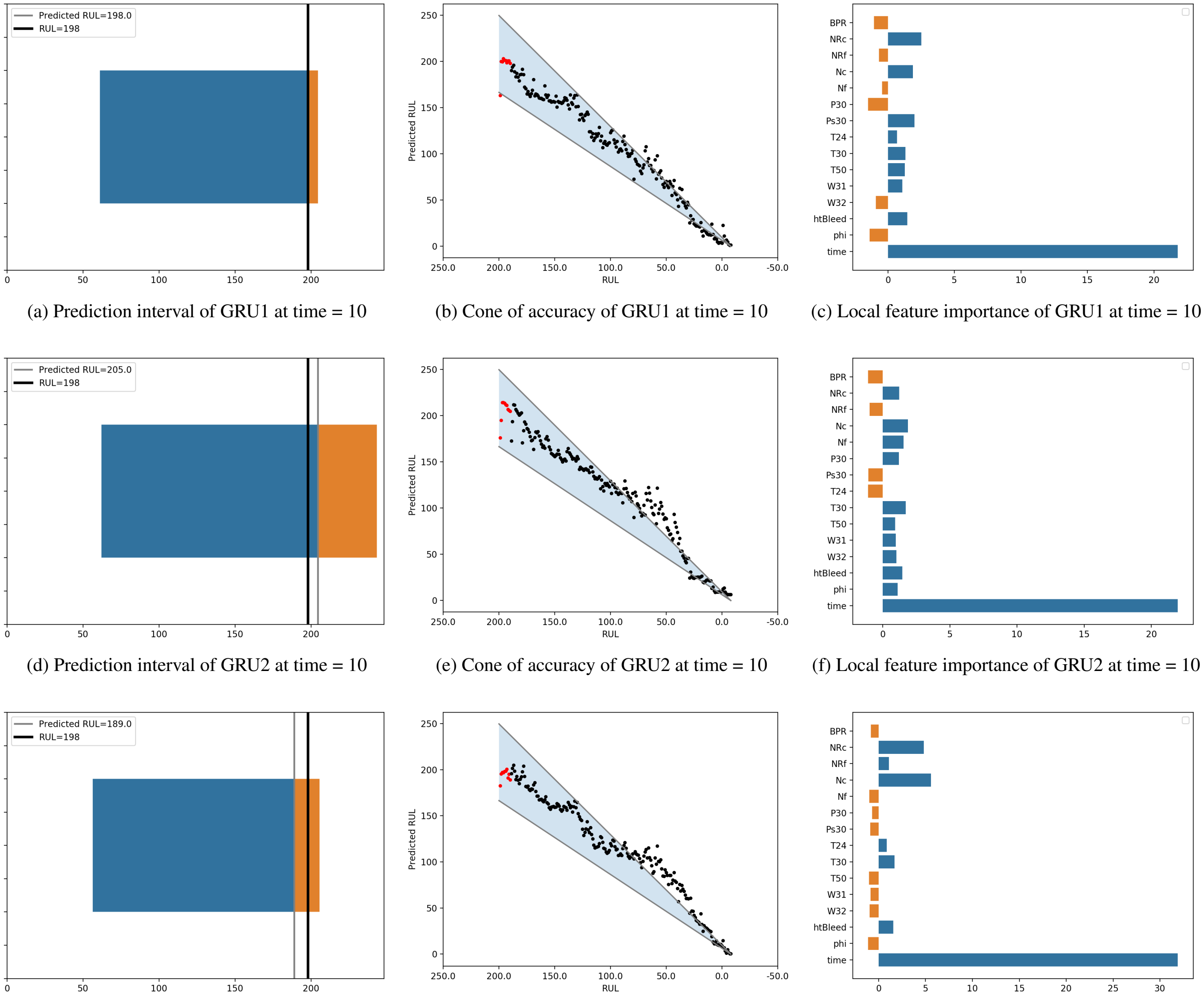
4.3. Local Interpretability with LIME

Although global interpretability is important for understanding the underlying mechanisms of prognostics, local interpretability can help the operator in her dedicated decision process. To analyze local interpretability in prognostics, we use the interval chart and the local feature importance chart. Examples of the interval chart are in Fig. 4a, 4d and 4g. This type of chart provides an idea of the interpretable search space G used to generate the LIME explanations $g \in G$. In other words, the interval chart shows the interval of perturbed decisions. It also shows the predicted RUL against the actual RUL.

Examples of the local feature importance chart are shown in Fig. 4c, 4f and 4i. This type of chart allows one to analyze the importance of local features with LIME. Here, we follow the convention of Ribeiro et al. (2016) where the height of the bar represents how much the feature contributes to the estimation. The positive bars, shown in blue, represent features that contribute to a large estimated value indicating we are far from the End Of Life (EoL). The negative bars, shown in orange, contribute towards moving closer to the EoL. In other words, positive bars are predictors of non-failure while negative bars are predictors of failure. We analyze the predictions of a single unit at different time instants ($t = 10, 90, 180$) in Fig. 4, 5 and 6. As shown, as we get closer to the EoL, the more orange bars we can see in the third column charts.

Fig. 4b, 4b and 4b illustrate the predictions of the three different GRU models applied to input instance at time $t = 10$. Here, GRU1 is the most accurate model (absolute error of zero) but the model that is most interpretable is GRU3 with an

[2] R^2 is a measure of how close data are to the fitted regression line.



(a) Prediction interval of GRU1 at time = 10

(b) Cone of accuracy of GRU1 at time = 10

(c) Local feature importance of GRU1 at time = 10

(d) Prediction interval of GRU2 at time = 10

(e) Cone of accuracy of GRU2 at time = 10

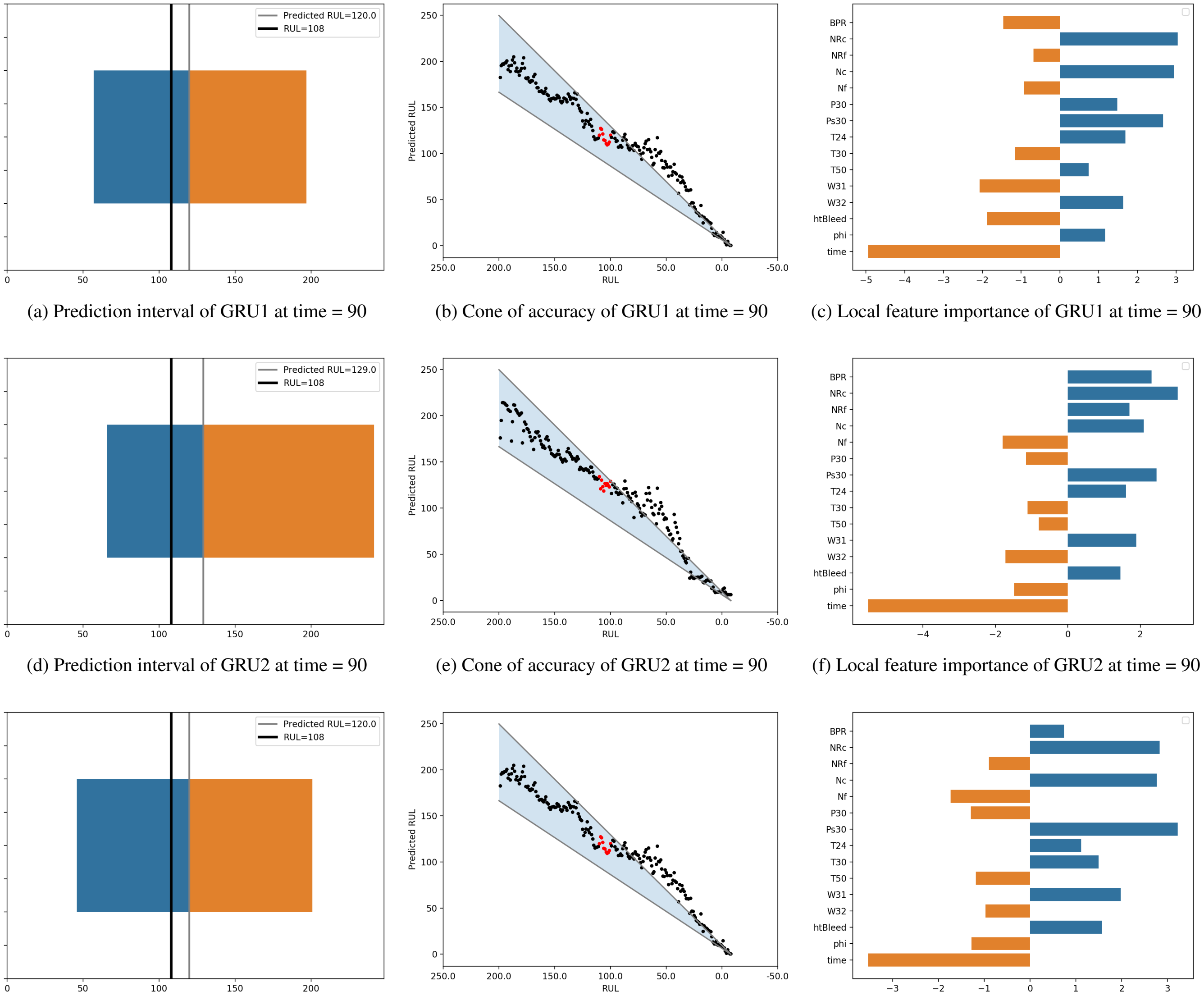
(f) Local feature importance of GRU2 at time = 10

(g) Prediction interval of GRU3 at time = 10

(h) Cone of accuracy of GRU3 at time = 10

(i) Local feature importance of GRU3 at time = 10

Figure 4. Local Interpretability of a decision made at time $t = 10$ for three GRU models. The fidelity of the explanation models measured in R^2 is 48.17, 23.98 and 60.54% for each GRU.



(a) Prediction interval of GRU1 at time = 90

(b) Cone of accuracy of GRU1 at time = 90

(c) Local feature importance of GRU1 at time = 90

(d) Prediction interval of GRU2 at time = 90

(e) Cone of accuracy of GRU2 at time = 90

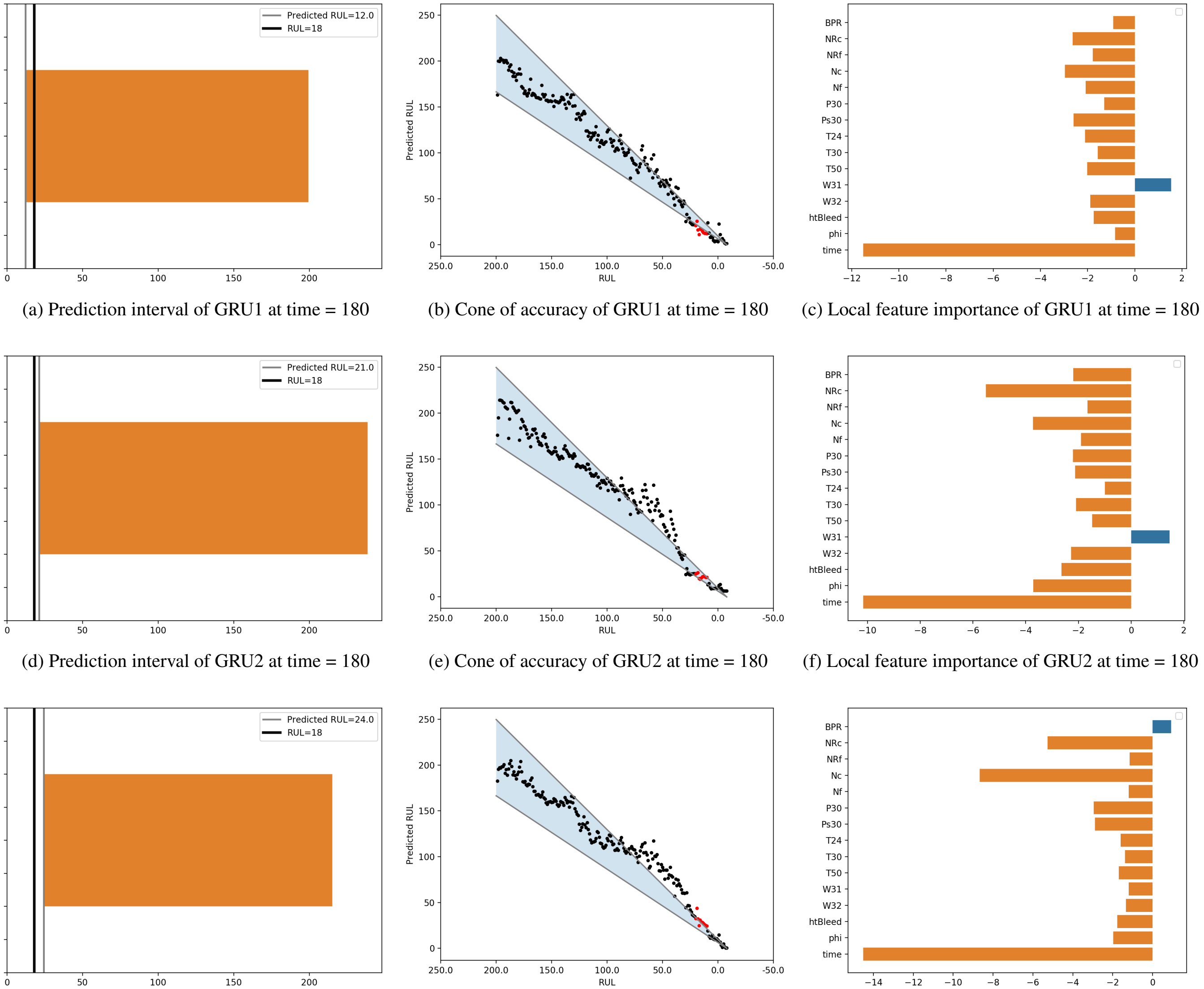
(f) Local feature importance of GRU2 at time = 90

(g) Prediction interval of GRU3 at time = 90

(h) Cone of accuracy of GRU3 at time = 90

(i) Local feature importance of GRU3 at time = 90

Figure 5. Local Interpretability of a decision made at time $t = 90$ for three GRU models. The fidelity of the explanation models measured in R^2 is 10.70, 9.14 and 10.17% for each GRU.



(a) Prediction interval of GRU1 at time = 180

(b) Cone of accuracy of GRU1 at time = 180

(c) Local feature importance of GRU1 at time = 180

(d) Prediction interval of GRU2 at time = 180

(e) Cone of accuracy of GRU2 at time = 180

(f) Local feature importance of GRU2 at time = 180

(g) Prediction interval of GRU3 at time = 180

(h) Cone of accuracy of GRU3 at time = 180

(i) Local feature importance of GRU3 at time = 180

Figure 6. Local Interpretability of a decision made at time $t = 180$ for three GRU models. The fidelity of the explanation models measured in R^2 is 16.50, 13.92 and 22.83% for each GRU.

R^2 of 60.54% versus the R^2 of 48.17% of GRU1 and 23.98% of GRU2 (see Table 4). This is despite GRU3 having the largest local error. This result reinforces the notion that accuracy is only one of the evaluation measures to consider in prognostics (Saxena, Celaya, et al., 2008).

Analyzing the feature importance charts of Fig. 4c, 4f and 4i, it can be seen that for all the considered GRUs, the results are consistent, in that there are more factors that indicate that we are far from the failure, i.e. more blue bars. If we consider the charts of Fig. 6c, 6f and 6i, in which we are closer to failure (time = 180), the situation is the opposite: we have more factors indicating failure, i.e. more orange bars. The charts of Fig. 5c, 5f and 5i depict how the factors indicate that we are at the half of the residual life of the equipment.

Interestingly, in Fig. 4i, the features *NRc* and *Nc* appear distinctively as the primary explanations to still be far from actual failure. These explanations were provided by the LIME model with the largest R^2 (highest fidelity) (see Tab. 4). Please note that these were also the features considered to be globally important (see Fig. 3b).

Analysis of the interval charts is also of significance. For instance in Fig. 4a, 4d and 4g, GRU2 has a small local error but is the least interpretable model (R^2 of 23.98%). In fact, the interval of perturbed decisions is the widest of all of the considered three. The R^2 is hence an important property of the explanation functions that should be considered when evaluating results.

As a final note, we analyze the results of Tab. 4. It can be seen that the local models may differ greatly in fidelity, i.e. in R^2 score. This result suggests that it is essential to understand the fidelity of the explanations in order to select the appropriate models.

5. CONCLUSION

To estimate the remaining useful life of a physical system or component is a complex and often decisive decision. In this paper we addressed this problem from the perspective of model interpretability. We study LIME, a locally interpretable model-agnostic approach and its ability to provide understanding about the prognostics of a commercial turbofan engine with the gated recurrent unit architecture. Our contribution is to advance the field of data-driven prognostics using interpretable models from eXplainable artificial intelligence to provide further insights into prognostics for better informed decision making.

In this work, we studied global and local interpretability using LIME. Even though LIME is a model originally designed for local interpretability we have shown that it can be used to obtain a better global understanding of a prognostics model. In addition, we have shown that the most important features according to the decision trees, the classical global interpretability method, may not necessarily be associated with the higher weights in LIME. This can be explained by the fact that certain characteristics, which can significantly influence individual decisions, may lose relevance between trees (Kazemitabar, Amini, Bloniarz, & Talwalkar, 2017). LIME can therefore be a way to obtain this important information.

Regarding local interpretability, the results suggest that it may be possible to trust local LIME models provided there is an examination of their fidelity and limitations. An interesting and consistent trend that we observed is that as we approach failure, the number of factors, and the intensity by which they signal failure, increases. The results also suggest that R^2 is an important trait to take into account when estimating the level of confidence to have in an explanation.

Future research directions include the use of new interpretability models such as DLIME (Zafar & Khan, 2019) or ALIME (Shankaranarayana & Runje, 2019) which try to address some of the limitations of LIME. Further studies on the consistency and stability of LIME results in prognostics are needed. This work intends to be an exploratory contribution to the field.

In this paper we evaluated LIME in a formal manner on a case study. It is also possible to informally evaluate interpretable models with maintenance staff in real-world scenarios. From this analysis it may be possible to assess if LIME explanations are correct (subjectively), and most importantly, if they are intuitive. This is another future research direction.

NOMENCLATURE

x	input sample
N	sampling size
M	number of explanation functions
x'	perturbed input sample
f	observed model
g	explanation model
g_L	best explanation local model
π_x	proximity measure
w_g	coefficients of linear model g
$\mathcal{L}$	fidelity function
Ω	penalty function
t	time step
T	time window size
EoL	End of Life
CBM	Condition-Based Maintenance
HPC	High Pressure Compressor
HPT	High Pressure Turbine
GRU	Gated Recurrent Unit
LIME	Local Interpretable Model-agnostic Explanations
LPC	Low Pressure Compressor
LPT	Low Pressure Turbine
LSTM	Long-Short Term Memory
PM	Predictive Maintenance
WLR	Weighted Linear Regression
RCM	Reliability Centered Maintenance
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
TBM	Time-Based Maintenance
XAI	eXplainable Artificial Intelligence

REFERENCES

Agarwal, R., Melnick, L., Frosst, N., Zhang, X., Lengerich, B., Caruana, R., & Hinton, G. E. (2021). Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, *34*, 4699–4711.

An, D., Kim, N. H., & Choi, J.-H. (2015). Practical options for selecting data-driven or physics-based prognostics algorithms with reviews. *Reliability Engineering & System Safety*, *133*, 223–236.

Antamis, T., Drosou, A., Vafeiadis, T., Nizamis, A., Ioannidis, D., & Tzovaras, D. (2024). Interpretability of deep neural networks: A review of methods, classification and hardware. *Neurocomputing*, 128204.

Autonomio Talos [Computer Software]. (2019).

Balasubramani, P., Shi, Q., & DeLaurentis, D. (2024). Explainable machine learning for turbojet engine prognostic health management. In *Aiaa scitech 2024 forum* (p. 0762).

Baptista, M. L., Goebel, K., & Henriques, E. M. (2022). Relation between prognostics predictor evaluation metrics and local interpretability shap values. *Artificial Intelligence*, *306*, 103667.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, *13*(Feb), 281–305.

Bonissone, P. P., & Goebel, K. (1999). Soft computing techniques for diagnostics and prognostics. In *Working notes for the 1999 aaai symposium for use of ai in equipment maintenance.*

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32.

Bulín, M., Šmídl, L., & Švec, J. (2019). On using stateful lstm networks for key-phrase detection. In *International conference on text, speech, and dialogue* (pp. 287–298).

Celaya, J. R., Saxena, A., & Goebel, K. (2012). *Uncertainty representation and interpretation inmmodel-based prognostics algorithms based on kalman filter estimation* (Tech. Rep.). NASA CA AMES RESEARCH.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078.*

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555.*

Daigle, M. (2014). *Model based prognostics.* PHM Society.

Dogga, B., Sathyan, A., & Cohen, K. (2024). Explainable ai based remaining useful life estimation of aircraft engines. In *Aiaa scitech 2024 forum* (p. 2530).

Dong, D., Li, X.-Y., & Sun, F.-Q. (2017). Life prediction of jet engines based on LSTM-recurrent neural networks. In *International conference of prognostics and system health management* (pp. 1–6).

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608.*

Erasmus, A., Brunet, T. D., & Fisher, E. (2021). What is interpretability? *Philosophy & Technology*, *34*(4), 833–862.

Fen, H., Song, K., Udell, M., Sun, Y., Zhang, Y., et al. (2019). Why should you trust my interpretation? understanding uncertainty in lime predictions. *arXiv preprint arXiv:1904.12991.*

Gawde, S., Patil, S., Kumar, S., Kamat, P., Kotecha, K., & Alfarhood, S. (2024). Explainable predictive maintenance of rotating machines using lime, shap, pdp, ice. *IEEE Access*, *12*, 29345–29361.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, *63*(1), 3–42.

Geurts, P., & Louppe, G. (2011). Learning to rank with extremely randomized trees. In *Jmlr: Workshop and conference proceedings* (Vol. 14, pp. 49–61).

Gunning, D., & Aha, D. (2019). Darpa's explainable artificial intelligence (xai) program. *AI magazine*, *40*(2), 44–58.

Hasib, A. A., Rahman, A., Khabir, M., & Shawon, M. T. R. (2023). An interpretable systematic review of machine learning models for predictive maintenance of aircraft engine. *arXiv preprint arXiv:2309.13310*.

Heimes, F. O. (2008). Recurrent neural networks for remaining useful life estimation. In *International conference on prognostics and health management* (pp. 1–6).

Hinchi, A. Z., & Tkiouat, M. (2018). A deep Long-Short-Term-Memory neural network for lithium-ion battery prognostics. In *International conference on industrial engineering and operations management* (pp. 2162–2168).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hu, X., Rudin, C., & Seltzer, M. (2019). Optimal sparse decision trees. In *Advances in neural information processing systems* (pp. 7265–7273).

Jaeger, H. (2001). *The "echo sttate" approach to analysing and training recurrent neural networks– With an erratum note* (Tech. Rep.). German National Research Center for Information Technology GMD Technical Report.

Jaeger, H. (2002). *Tutorial on training recurrent neural networks, covering bppt, rtrl, ekf and the "echo state network" approach* (Vol. 5). GMD-Forschungszentrum Informationstechnik Bonn.

Jardine, A. K., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, *20*(7), 1483–1510.

Ji, Z., Zhang, L., & Yan, W. (2024). An interpretable fault prediction method based on machine learning and knowledge graphs. In *International conference on intelligent computing* (pp. 30–41).

Kazemitabar, J., Amini, A., Bloniarz, A., & Talwalkar, A. S. (2017). Variable importance using decision trees. In *Advances in neural information processing systems* (pp. 426–435).

Kim, B. (2015). *Interactive and interpretable machine learning models for human machine collaboration* (Unpublished doctoral dissertation). Massachusetts Institute of Technology.

Kulkarni, C. S., Daigle, M. J., Gorospe, G., & Goebel, K. (2018). Experimental validation of model-based prognostics for pneumatic valves.

Kundu, R. K., & Hoque, K. A. (2023). Explainable predictive maintenance is not enough: quantifying trust in remaining useful life estimation. In *Annual conference of the phm society* (Vol. 15).

Lakkaraju, H., Bach, S. H., & Leskovec, J. (2016). Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1675–1684).

Lee, J., Holgado, M., Kao, H.-A., & Macchi, M. (2014). New thinking paradigm for maintenance innovation design. *IFAC Proceedings Volumes*, *47*(3), 7104–7109.

Lee, K., Kim, J.-K., Kim, J., Hur, K., & Kim, H. (2018a). CNN and GRU combination scheme for bearing anomaly detection in rotating machinery health monitoring. In *1st international conference on knowledge innovation and invention (ickii)* (pp. 102–105).

Lee, K., Kim, J.-K., Kim, J., Hur, K., & Kim, H. (2018b). Stacked convolutional bidirectional LSTM recurrent neural network for bearing anomaly detection in rotating machinery diagnostics. In *1st international conference on knowledge innovation and invention (ickii)* (pp. 98–101).

Lipton, Z. C. (2018). The mythos of model interpretability. *Queue*, *16*(3), 31–57.

Litt, J. S., Frederick, D. K., & DeCastro, J. (2008). *Simulating operation of a large turbofan engine* (Tech. Rep.). NASA.

Long, B., Li, X., Gao, X., & Liu, Z. (2019). Prognostics comparison of lithium-ion battery based on the shallow and deep neural networks model. *Energies*, *12*(17), 1–12.

Lou, Y., Caruana, R., & Gehrke, J. (2012). Intelligible models for classification and regression. In *Proceedings of the 18th acm sigkdd international conference on knowledge discovery and data mining* (pp. 150–158).

Ma, R., Yang, T., Breaz, E., Li, Z., Briois, P., & Gao, F. (2018). Data-driven proton exchange membrane fuel cell degradation predication through deep learning method. *Applied Energy*, *231*, 102–115.

Melis, D. A., & Jaakkola, T. (2018). Towards robust interpretability with self-explaining neural networks. In *Advances in neural information processing systems* (pp. 7775–7784).

Morgan, N., & Bourlard, H. (1990). Generalization and parameter estimation in feedforward nets: Some experiments. In *Advances in neural information processing systems* (pp. 630–637).

Moubray, J. (2001). *Reliability-centered maintenance*. Industrial Press Inc.

NASA, RCM. (2008). Guide reliability: Centered maintenance guide for facilities and collateral equipment. *Aeronautics and SA NASA, Eds.*

Nguyen, D., Kefalas, M., Yang, K., Apostolidis, A., Olhofer, M., Limmer, S., & Bäck, T. (2019). A review: Prognostics and health management in automotive and aerospace. *International Journal of Prognostics and Health Management*, *10*(2), 35.

Quinlan, J. R. (1986). Induction of decision trees. *Machine* *learning*, *1*(1), 81–106.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “why should i trust you?” explaining the predictions of any classifier. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 1135–1144).

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for evaluating performance of prognostic techniques. In *International conference on prognostics and health management* (pp. 1–17).

Saxena, A., Goebel, K., Simon, D., & Eklund, N. (2008). Damage propagation modeling for aircraft engine run-to-failure simulation. In *International conference on prognostics and health management* (pp. 1–9).

Schoenborn, J. M., & Althoff, K.-D. (2019). Recent trends in xai: A broad overview on current approaches, methodologies and interactions. In *Case-based reasoning for the explanation of intelligent systems (xcbr) workshop.*

Schwabacher, M., & Goebel, K. (2007). A survey of artificial intelligence for prognostics. In *Aaai fall symposium: Artificial intelligence for prognostics* (pp. 108–115).

Serradilla, O., Zugasti, E., Cernuda, C., Aranburu, A., de Okariz, J. R., & Zurutuza, U. (2020). Interpreting remaining useful life estimations combining explainable artificial intelligence and domain knowledge in industrial machinery. In *2020 ieee international conference on fuzzy systems (fuzz-ieee)* (pp. 1–8).

Shankaranarayana, S. M., & Runje, D. (2019). Alime: Autoencoder based approach for local interpretability. In *International conference on intelligent data engineering and automated learning* (pp. 454–463).

Song, Y., Li, L., Peng, Y., & Liu, D. (2018). Lithium-ion battery remaining useful life prediction based on grurnn. In *2018 12th international conference on reliability, maintainability, and safety (icrms)* (pp. 317–322).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Tse, P., & Atherton, D. (1999). Prediction of machine deterioration using vibration based fault trends and recurrent neural networks. *Journal of Vibration and Acoustics*, *121*(3), 355–362.

Turbé, H., Bjelogrlic, M., Lovis, C., & Mengaldo, G. (2023). Evaluation of post-hoc interpretability methods in time-series classification. *Nature Machine Intelligence*, *5*(3), 250–260.

Tzeng, F.-Y., & Ma, K.-L. (2005). *Opening the black box-data driven visualization of neural networks*. IEEE.

Ucar, A., Karakose, M., & Kırımça, N. (2024). Artificial intelligence for predictive maintenance applications: key components, trustworthiness, and future trends. *Applied Sciences*, *14*(2), 898.

Yuan, M., Wu, Y., & Lin, L. (2016). Fault diagnosis and remaining useful life estimation of aero engine using lstm neural network. In *International conference on aircraft utility systems (aus)* (pp. 135–140).

Zafar, M. R., & Khan, N. M. (2019). DLIME: a deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems. *arXiv preprint arXiv:1906.10263*.

Zeldam, S. (2018). *Automated failure diagnosis in aviation maintenance using explainable artificial intelligence (xai)* (Unpublished master’s thesis). University of Twente.

Zhang, W., Jin, F., Zhang, G., Zhao, B., & Hou, Y. (2019). Aero-engine remaining useful life estimation based on 1-dimensional FCN-LSTM neural networks. In *Chinese control conference (ccc)* (pp. 4913–4918).

Zhang, Y., Xiong, R., He, H., & Liu, Z. (2017). A lstm-rnn method for the lithuim-ion battery remaining useful life prediction. In *International conference on prognostics and system health management* (pp. 1–4).

BIOGRAPHIES

Marcia L. Baptista (BS and MSc. in Informatics and Computer Engineering. Instituto Superior Tecnico, Lisbon, Portugal) holds a PhD from the Engineering Design and Advanced Manufacturing (EDAM) program under the umbrella of MIT Portugal. Her research focuses on the development of prognostic techniques for aerospace equipment. Her interests include data-driven modeling, prognostics, and deep learning.

Madhav Mishra holds a PhD degree in operation and maintenance Engineering from Luleå University of Technology (LTU), Sweden, and a Master’s degree in Control Systems Engineering with a specialization in Mechatronics from the Netherlands. He is currently a Senior Scientist at RISE Research Institutes of Sweden, where he leads research at the intersection of AI/ML, power electronics, predictive maintenance, and enhanced reliability for electronic systems.

Elsa Maria Pires Henriques has a doctorate in Mechanical Engineering and is associated professor at Instituto Superior Tecnico in the University of Lisbon. She is responsible for the “Engineering Design and Advanced Manufacturing (LTI/EDAM)” post-graduation. During the last fifteen years, she has participated and/or coordinated several national and European R&D projects in collaboration with different industrial sectors, from tooling to automotive and aeronautics, mainly related to manufacturing, life cycle based decisions and management of complex design processes. She has a large number of scientific and technical publications in national and international conferences and journals. She was a national delegate in the 7th Framework Programme of EU.

Helmut Prendinger received his Master and Doctoral degrees in Logic and Artificial Intelligence from the University of Salzburg in 1994 and 1998, respectively. Since 2012, he is a full professor at the National Institute of Informatics (NII), Tokyo, after joining NII in 2004 as Associate Professor. Previously, he held positions as a research associate (2000 – 2004) and JSPS postdoctoral fellow (1998 – 2000) at the University of Tokyo, Dept. of Information and Communication Engineering, Faculty of Engineering. In 1996-1997, he was a junior specialist at the University of California, Irvine. His research interests include artificial intelligence including machine learning, intelligent user interface, cyber-physical systems, and the melding of real and virtual worlds, in which areas he has published more than 220 peer-reviewed journal and conference papers. His vision is to apply his research to establish the IT infrastructure for Unmanned Aerial Vehicles, or “drone”. He is a member of IEEE and ACM.