

Deep Recurrent Entropy Adaptive Model for System Reliability Monitoring

Miguel Martínez-García, *Member, IEEE*, Yu Zhang, *Member, IEEE*, Kenji Suzuki, *Senior Member, IEEE*, and Yu-Dong Zhang, *Senior Member, IEEE*

Abstract—The aim of this paper is to develop a methodology for measuring the *degree of unpredictability* in dynamical systems with memory, i.e., systems with responses dependent on a history of past states. The proposed model is generic, and can be employed in a variety of settings, although its applicability here is examined in the particular context of an industrial environment: gas turbine engines. The given approach consists in approximating the probability distribution of the outputs of a system with a deep recurrent neural network; such networks are capable of exploiting the memory in the system for enhanced forecasting capability. Once the probability distribution is retrieved, the *entropy* or *missing information* about the underlying process is computed, which is interpreted as the uncertainty with respect to the system's behaviour. Hence the model identifies how far the system dynamics are from its typical response, in order to evaluate the system reliability and to predict system faults and/or *normal accidents*. The validity of the model is verified with sensor data recorded from commissioning gas turbines, belonging to normal and faulty conditions.

Index Terms—Entropy, Recurrent neural networks, Deep learning, Condition monitoring, Long short-term memory, Cognitive Computing

I. INTRODUCTION

AS technology advances, the complexity of the physical environment around us grows at an increasing rate. This is especially conventional in complex industrial environments, where the expansion of the so-called Industry 4.0 is rising the degree of interactivity between the ever larger number of subsystems. *Complex systems* typically display dynamics that can only be modelled through computationally expensive approaches [1], and do not allow for reduction to simple systems of equations. This is a result of their dynamics being produced by the interaction of their many subsystems – what is known in the complexity theory as *emergent phenomena*. Thus in a complex system, subsets of its subsystems cannot explain the global *emergent* dynamics, making prediction tasks particularly challenging.

C. Perrow studied the issue of low system predictability within the setting of industrial plants. In his influential work [2], he referred to these systems with the descriptive term of *highly coupled systems*. In the same publication, numerous examples of low probability and low predictability *accidents* or

system faults are given. These accidents are termed as *normal accidents* because, although uncommon, they are bound to occur in the long run. For the reason that the unpredictability of such systems is a consequence of subsystems interaction, adding automatic safety mechanisms may further increase their complexity [2], potentially generating new types of *normal accidents*. Hence, the approach proposed in this paper consists in identifying abnormal behaviour in a system, instead of characterizing particular faulty conditions; there are far too many potential faults, some of which can be regarded as *normal accidents*. In many occasions these are originated by concomitant causes, like control systems failure, human error or sensor misreadings (Sec III-A).

The method studied here utilizes a measure to estimate unpredictability or *missing information* with respect to a system. When the unpredictability of the system's state variables increases, the behaviour of the system is regarded as anomalous. For such a task, an uncertainty measure known as *information entropy* has long been established [3]. Indeed, when Shannon proposed this measure, he also proved that it is the only possible formulation that satisfies the three basic reasonable requirements that any measure of information should comply [4]. The entropy measure is a function that takes as input a probability function, and yields as output a value indicating how much it is unknown about the underlying generating process; it is a measure of the amount of missing information with respect to the system at hand [4].

Lately, entropy analysis has received substantial attention in the domain of biomedical signal analysis [5], [6], and a fair level of attention for condition monitoring of industrial systems [7], [8]. A number of modified entropy measures are surveyed in [9]. As stated, any alternative entropy formulation must satisfy the three basic requirements described by Shannon [4]. Hence generally, these newly proposed measures are reformulations of information entropy over different probability functions.

Since entropy measurement depends on a probability function, any entropy based condition monitoring method needs an approach to estimate it. This can be done in various ways. One common scheme is direct approximation from the response signals of the system [10]. Nonetheless, when sequential observations in a time series are not independent, this may result in a biased estimate of the probability function. Another method is to use a model to predict the state of the system. For example, the errors between the model output and the actual system values can be treated as samples of a random variable. This strategy has been used successfully in [11] as a measure of driver's workload level, and later applied to calibrate steering

Miguel Martínez García (e-mail: M.Martinez-Garcia@lboro.ac.uk) and Yu Zhang are with the Department of Aeronautical and Automotive Engineering, Loughborough University, U.K.

Kenji Suzuki is with the Institute of Innovative Research, Tokyo Institute of Technology, Japan

Yu-Dong Zhang is with the Department of Informatics, University of Leicester, Leicester, U.K.

control models [12]. Here an alternative method is utilized, in which a model – by way of an Artificial Neural Network (ANN) – yields directly as output a probability function.

Generally, system dynamics in industrial environments can be modelled by linear systems of differential equations. But linear approximations are only accurate in individual and restricted operational regimes. In most cases, system dynamics are better characterized by non-linear equations or by fractional operators with memory [13], [14]. All these methods are even more relevant when the response of the system examined diverges from its typical behaviour; when higher and unpredictable subsystem interactivity may bifurcate its response towards chaotic dynamics or dynamics exhibiting larger hysteresis [15].

ANNs are appropriate for modelling linear as well as non-linear dynamics [16], but their classical formulation does not accommodate for memory properties. Nonetheless, for sequential data analysis, Recurrent Neural Networks (RNN) can be employed. RNNs are a specialized type of ANN that maintain context – or memory – when trained with sequential data [17]. Examples of its applicability include weather forecast [18], stock market prediction [19] and machine translation [20]. One pertinent aspect of any ANN is that, when trained as a classifier, it approximates not only the forecasted output but also its probability distribution. Hence in this paper a RNN will be employed to approximate a probability distribution, with which the entropy measure can be computed. And this will be done with the aim of characterizing system predictability, as a way of identifying anomalous system behaviour.

Further, to provide a case study for validation purposes Industrial Gas Turbines (IGTs) are considered. IGTs are a true example of complex systems organized in many subsystems: compressor, combustors, pumps, fuel supply, ignition system, lubrication and drain modules, etc [21]. And as complex machines, IGTs can display a wide-ranging repertoire of anomalous behaviour (Sec. III-A). The proposed methodology is here validated for the particular case of condition monitoring of IGTs. Nevertheless, it has general applicability to other systems from varying domains, such as biomedical data analysis and any *high risk technology* generating sequential data or sensor data from arrays of sensor networks [22], [23]. A review of other information fusion methods for IGT diagnostics can be found in [24].

The remainder of paper is organized into the following sections: in Section II the required background elements of information theory and RNNs are introduced. Next, in Section III, the characteristics of the used data are specified. Section IV describes the proposed model and the parameter fitting procedure, while in Section V the model is tested and validated with real-world sensor data recorded from IGTs. Discussion and conclusions are found in Section VI.

II. UNDERLYING PRINCIPLES

A. Entropy or Missing Information

Although the concept of entropy was originated in the study of thermodynamics, today it is often used in multiple fields that are not related to energy transfer, e.g. language analysis [25] and redundancy estimation in the genome [26]. Most of

these usages emerge from the information theory interpretation of entropy, which is the topic examined here.

Information entropy (or entropy), is a measure of the amount of missing information with respect to a process, understood as a random variable. Given an information source modelled by a random variable X , for which a probability distribution $P(X)$ is known, and assuming that X has $n \in \mathbb{N}$ possible outcomes, the entropy $H_\beta(X)$ is defined as:

$$H_\beta(X) := - \sum_{i=1}^n p_i \log_\beta p_i, \quad (1)$$

where $p_i = P(X = x_i)$ for $i \in \{1 \dots n\}$. When $\beta = 2$, the units of H_2 are bits.

$H_\beta(X)$ does not depend on the underlying process X being divided into parts, as that does not change the amount of uncertainty about the process [4]. This property reflects a very important idea; the missing information about a system is equal to the missing information of each of its subsystems plus the missing information corresponding to subsystem interaction – i.e., the emerging complexity in a tightly coupled system [2].

A descriptive way to understand the definition in (1), is by comparing it to classical statistical measures. Consider two random variables – both with uniform probabilities: $X_1 = \{0, 1\}$ and $X_2 = \{0, 0.25, 0.75, 1\}$ (Fig. 1). X_2 has a higher number of equally likely outcomes than X_1 , therefore $H_2(X_2) > H_2(X_1)$ (X_2 is less predictable and less reliable). Contrarily, for the standard deviation $\sigma(X_1) > \sigma(X_2)$. Simultaneously, both random variables exhibit the same expected value $E(X_1) = E(X_2) = 1/2$. H_β is a measure about the potentiality of a system (Fig. 2), and not a static simplification like mean and variance.

With respect to condition monitoring techniques, although machine learning algorithms based on classical statistics have been successfully applied for particular operational regimes [27], they are challenged when hardware-controlled or environmental variations dominate the sensor data. Diversely, entropy based techniques are able to characterize how little it is known about the system, irrespective of the summary statistics [10].

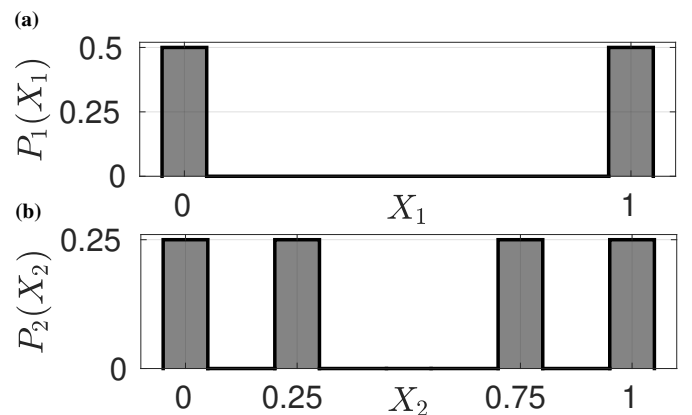


Fig. 1: The probability function $P_1(X_1)$ in (a) has a higher variance ($\sigma^2 = 0.25$) than that in (b) ($\sigma^2 \approx 0.16$). On the other hand, $P_2(X_2)$ in (b) has a higher entropy ($H_2(X_2) = 2$) than $P_1(X_1)$ in (a) ($H_2(X_1) = 1$). In both cases $E(X_{1,2}) = 0.5$.

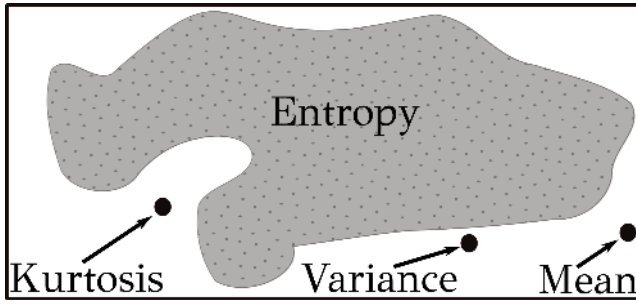


Fig. 2: While summary statistics give information about the minority of elements that can be known about a stochastic process, H_β shows how much it is not known – informally it yields the size of the unknown (shaded gray area).

Increased entropy can indicate system anomalies in different operational regimes – such as varying system workload.

B. Recurrent Neural Networks

ANNs and information entropy originated from participants of the influential Macy Conferences on cybernetics (1941–60) [28], which established principal foundations towards interdisciplinary science. Today, both computational approaches are deeply intertwined, as ANNs employed as classifiers are often trained by means of an entropy loss function (see Eq. 2).

ANNs allow for the modelling of complex systems without the need of extensive domain knowledge, and have been used for condition monitoring purposes through supervised [7], [29] and unsupervised approaches [8], [30]. When the input features are sequential in nature, one common approach is to use recurrent models with a delayed feedback loop in each layer, i.e., RNNs. Each layer in a RNN includes a memory block that maintains information across an arbitrary number of iterative steps.

The first ANNs with recurrent properties can be traced back to the 1980s [31], but it was not until much later that models, robust to the *vanishing gradients problem*, were developed: notably, the Long Short-term Memory (LSTM) model [17] and models based on Gated Recurrent Units (GRUs) [32]. In both architectures, the degree of memory retention is controlled by a gating mechanism. Although the relative efficacy between these methods is still under debate [33], LSTM models seem more suitable when large data are available. Thus in the present paper the LSTM architecture is adopted. Novel applications of LSTM cells to model complex systems include: medical diagnosis [34] and monitoring superconducting magnets [35], [36]. Several improvements of the original LSTM model can be found in the literature. Here it is considered the LSTM implementation incorporated in the TensorFlow library [37], [38].

In the literature different optimization algorithms have been proposed to train the ANN. These range from the Levenberg-Marquardt methods [39] to the Widrow-Hoff rule [40]. RNNs are typically trained by using a variant of the standard backpropagation algorithm, which is named *backpropagation through time* (BPTT). The choice of loss function depends on the particular problem. For regression problems the Mean

Squared Error (MSE) can be employed, while for classification tasks the *cross-entropy* measure is more suitable. The latter is defined as:

$$H_2(p, q) := - \sum_{i=1}^n p_i \log_2 q_i \quad (2)$$

where p_i are the true probabilities inferred from the labels in the training data, and q_i are approximated by the model through a *softmax* layer stacked after the recurrent layer. Additionally, when sufficient data are available, it is possible to augment the effectiveness by stacking one recurrent layer after another – leading to a *deep learning* model (Fig. 4).

III. EXPERIMENTAL DATA

A. Elements of Industrial Gas Turbines

In this paper, sensor data recorded from IGT engines are employed (Fig. 3a). Essentially an IGT engine transforms fuel energy into shaft power. When the engine is in operation, shaft rotation causes the rotor blades to turn into the air compressor. The rotors action induces the intake of atmospheric air from the compressor inlet (Fig. 3b), and its subsequent pressurization as it flows through the various rotor and stator blades, which are sequentially spaced within the compressor. The resulting high pressure air is then released into the combustors, where the temperature of the compressed air is highly increased by burning liquid or gas fuel. Subsequently, the expansion produced by the high pressure and temperature causes the air to expand towards the power generator part of the engine, which consists of another series of rotors connected to the same shaft. A part of the generated shaft power is employed to compress new atmospheric air via a self-sustaining mechanism, which was initiated by an auxiliary electric motor. The new air prevents the flow from reverting backwards and stalling the engine.

There are two types of IGT engines. In the first kind, the residual air from the power generator is directly pushed out through the exhaust as new air comes in. These engines have a single shaft and are typically employed for electrical power generation. Unlike *single-shaft* engines, *twin-shaft* engines have a supplementary shaft, immediately after but decoupled from the first shaft. This mechanism yields more control on the applied shaft torque. Thus twin-shaft engines, such as the ones investigated here, are generally used for mechanical power generation – for example, on offshore drilling rigs.

B. Data Characteristics

The data were recorded from 10 twin-shaft IGT engines, situated in different geographical locations with varying environmental conditions. The dataset was prepared by Siemens to analyze the performance and to monitor the condition of their machines. Typically a customer can choose to install a number of sensors on an IGT. The raw data from these sensors are recorded in real time and stored by the OEM for future use, without any preprocessing or filtering. The majority of the data correspond to IGTs in healthy operating conditions. The dataset also includes message logs, announced by the control system – such as warnings and automatic shutdowns – and site

visit reports written by the operators. These allow to identify segments of data corresponding to a faulty machine. The data from healthy machines were used to train the model (Sec. IV), while the samples from faulty machines were employed for the case studies (Sec. V). Further information of the dataset, such as the geographical location of the engines, the exact position of the sensors and their specifications is confidential.

Measurements from 35 sensors were employed in this study; these coincide with health indicators known to be correlated to different IGT component conditions [27] (Tab. I). The sampling period was of 1 min, and comprised several months of data collected from each engine. Besides the data corresponding to normal engine functionality, data from different faulty conditions were collected (see Sec. V). Prior to their use for RNN training, the data were curated; data segments with missing sensors were discarded and the data were standardized¹ (Sec. IV-B). The curated data were split into training, development and testing sets (see Tab. II).

IV. ANN MODEL ARCHITECTURE

In this section, two RNN models are described: a regressor and a classifier. The regressor is applied for demonstrating the feasibility of the methodology and to select the appropriate RNN architecture. The classifier, referred to as the *Deep Recurrent Entropy Adaptive Model* (DREAM), is an extension

¹To zero mean and unit variance.

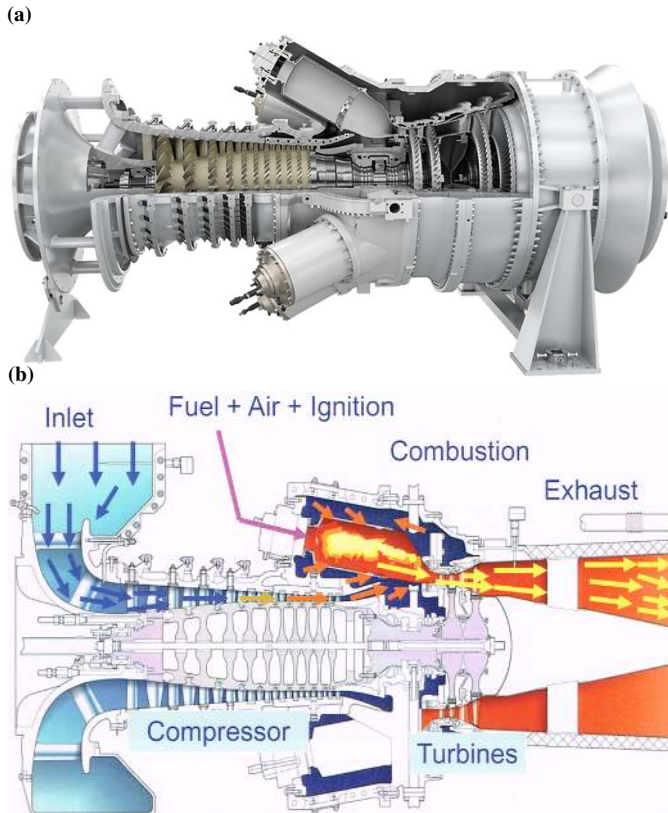


Fig. 3: (a) Visual depiction of a *twin-shaft* IGT engine. (b) Diagram of the basic operation flow in a *single-shaft* IGT engine.

of the regressor, and is able to approximate the probability distribution of the outputs of a system and compute its entropy. The DREAM is used as an algorithm for system unreliability evaluation (Sec. V).

A. Deep Regressor RNN model

The regressor RNN is designed to predict the shaft power produced by an IGT (*S-35* in Tab. I) at time step $< t >$, from the inputs of the sensors $S-01, \dots, 34$ within a window of length L_W : $\{< t > - L_W + 1, \dots, < t >\}$. Thus the model uses as input the last L_W observations of the states of the machine. The last layer of the model is fully connected – i.e., a *dense layer*, to generate a one-dimensional output: the predicted standardized shaft power. The regressor was optimized by minimizing the MSE from the training data through BPTT, with the Adaptive Moment (Adam) algorithm. Hyperparameter tuning was conducted by *random search* [41]. This approach is known to be more effective than grid search, as for the same number of evaluations more values of each particular hyperparameter are tested. In total the RNN was trained 50 times with different hyperparameter combinations. The combination that resulted in the lowest error in the development set (Tab. II) is the one reported in the paper. The optimized hyperparameters include the window length, Adam's learning rate, number of cells per LSTM layer and number of LSTM layers. The retention probability of a *dropout* regularization method [42] was also tuned, which acted by randomly switching off a percentage of the inputs at each layer during the training. The tuned hyperparameters are shown in Table II. Longer or shorter memory windows than $L_W = 5$ resulted in reduced RNN performance, but this value is highly dependent on the sampling rate.

For the examined data, a two LSTM layer model conveys the best performance. Dropout regularization is not significantly effective in this case, possibly because the degree of overfitting is low for the selected number of epochs – which was selected through the *early-stopping* method; more epochs decreased performance on the development set. Nevertheless, as it makes the model more robust to sensor misreadings, it is retained in the model. For the final hyperparameter selection (Table II),

S#	SENSOR MEASUREMENT	UNIT
S-01	Fuel Power	kW
S-02	Inlet air temperature	°C
S-03	Inlet air pressure	Bar
S-04	Compressed air temperature	°C
S-05	Compressed air pressure	Bar
S-06,...,11	Combustors temperature (1-6)	°C
S-12,...,15	Combustors vibration (1-4)	Hz
S-16,...,28	Interduct temperatures (1-13)	°C
S-29	Shaft displacement – <i>x</i> axis	μm
S-30	Shaft displacement – <i>y</i> axis	μm
S-31, 32	Shaft temperatures (1-2)	°C
S-33	Shaft Speed	RPM
S-34	Exhaust air temperature	°C
S-35	Shaft Power	kW

TABLE I: Sensor data, recorded from the IGT engines used in this study.

the respective MSE over the training, development and testing sets are 0.0069, 0.0084 and 0.0103.

SET	#OF INSTANCES	%
Training	2 115 232	80%
Development	396 606	15%
Testing	132 202	5%
SYMBOL	HYPERPARAMETER	VALUE
l_r	Learning rate	0.0005
N_{epoch}	Number of epochs	90
L_W	Window length	5
N_L	Number of LSTM layers	2
N_c^1	Number of cells in layer 1	70
N_c^2	Number of cells in layer 2	30
P_{keep}	Dropout keep probability	0.99
N_σ	Number of bins (DREAM)	10

TABLE II: Data-set split from an initial set of 2 644 040 instances, along with the hyperparameter selection for the regressor model and the DREAM.

B. Deep Recurrent Entropy Adaptive Model

Based on the regressor model (Sec. IV-A), a classifier is constructed. In the DREAM, the fully connected layer, with one-dimensional output, is replaced by a ten-dimensional softmax layer, i.e., a fully connected ten-dimensional output layer (with outputs $y^{<t>,1}, y^{<t>,2}, \dots, y^{<t>,10}$) followed by a softmax activation function:

$$\sigma(y^{<t>,j}) = \frac{e^{y^{<t>,j}}}{\sum_{i=1}^{10} e^{y^{<t>,i}}}. \quad (3)$$

To train the DREAM, the standardized output (from Sensor S-35) was converted to a categorical variable. This was done by binning the numerical values into 10 classes (Tab. III), where classes 2 to 9 cover two standard deviations away from the mean in each direction ($\approx 95\%$ of the data). This bucketing strategy represents a range of arbitrary width within a finite number of classes, which was 10 in this case – this number keeps the number of RNN parameters to a minimum while yielding acceptable resolution. Sensor signals from IGTs under constant load display fluctuations, due to instantaneous variability of the dynamics and measurement noise. The number of buckets was chosen so that the corresponding signals stay within the same bucket while the machine is kept at constant load. This effect can be observed by comparing Figs. 5 and 6, where the signal of an IGT is displayed for a period of 21 days during which the load was altered multiple times. The central bins (2 to 9) embed the typical IGT dynamics, while the boundary bins (1 and 10) characterize outlier behaviour.

Class	1	2	3	4	5
Range σ	$(-\infty, -2]$	$(-2, -1.5]$	$(-1.5, -1]$	$(-1, -0.5]$	$(-0.5, 0]$
Class	6	7	8	9	10
Range σ	$(0, 0.5]$	$(0.5, 1]$	$(1, 1.5]$	$(1.5, 2]$	$(2, \infty)$

TABLE III: Bucketing of the standardized output from sensor S-35 into ten different classes – according to the standard deviation ($\sigma = 1$).

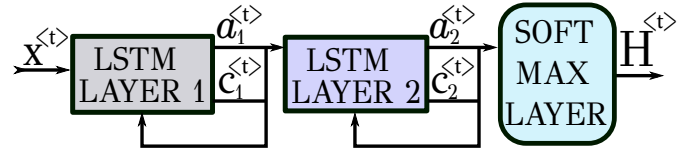


Fig. 4: Computational graph of the DREAM unpredictability measure, consisting in this case of two LSTM layers and a softmax layer. From the output of the softmax layer the entropy $H^{<t>}$ is computed.

The loss function is the cross-entropy (2), where the *true* conditional probabilities are given by the corresponding class; for a particular training set input sample $x^{<t>} \in \mathbb{R}^{34}$ with class label k

$$\tilde{y}^{<t>} = k \in \{1, \dots, 10\}, \quad (4)$$

the assigned conditional probability function is:

$$\begin{cases} p_k^{<t>} = P(\tilde{y}^{<t>} = k | x^{<0>, \dots, x^{<t>}}) = 1 \\ p_j^{<t>} = P(\tilde{y}^{<t>} = j | x^{<0>, \dots, x^{<t>}}) = 0 \quad \text{if } j \neq k. \end{cases} \quad (5)$$

The approximated probability function is estimated by the RNN from Eq. (3): $q_j^{<t>} = \sigma(y^{<t>,j})$, satisfying $\sum_{j=1}^{10} q_j^{<t>} = 1$.

The fundamental element of the DREAM relies on the following property of the entropy measure [4]:

$$\begin{aligned} H_\beta^{<t>}(p) &= - \sum_{i=1}^n p_i^{<t>} \log_\beta p_i^{<t>} \leq \\ &- \sum_{i=1}^n p_i^{<t>} \log_\beta q_i^{<t>} = H_\beta^{<t>}(p, q). \end{aligned} \quad (6)$$

Specifically, the cross-entropy is always greater than or equal to the entropy, and only equal when $p_j^{<t>} = q_j^{<t>}$ for all j . Accordingly, as the optimization process minimizes the cross-entropy, it approximates the *true* conditional probability function of the output variable for a particular input, and consequently its entropy:

$$\lim_{<t> \rightarrow \infty} q_j^{<t>} = p_j^{<t>}, \quad \forall j \in \{1, \dots, 10\}, \quad (7)$$

where the limit denotes the ideal case of unrestricted training data – representative of the underlying dynamics – over an idealized recurrent model, with capacity to learn fully the conditional probabilities in the data.

The DREAM was trained with the same hyperparameter set as in the regressor model (Tab. II), as it was found that further hyperparameter tuning does not provide significant difference in performance. The respective values of the cross-entropy loss over the training, development and testing sets are 0.036, 0.059 and 0.082.

The computational graph of the model is shown in Fig. 4. The trained model, when used for feed-forward prediction, contains an additional step to compute the entropy. For simpler interpretation, the entropy is normalized $H^{<t>} \in [0, 1]$:

$$H^{<t>} := H_{10}^{<t>}(q) = - \sum_{i=1}^{10} q_i^{<t>} \log_{10} q_i^{<t>}. \quad (8)$$

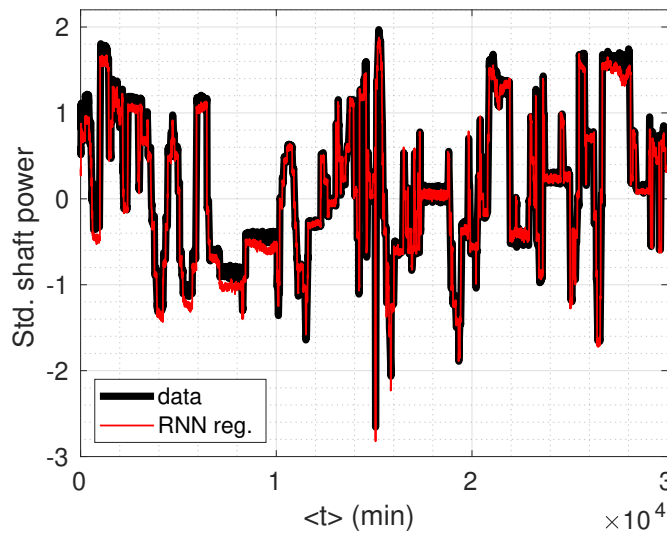


Fig. 5: Standardized shaft power produced by a healthy IGT engine during approximately 21 days and output of the deep RNN regressor model.

V. CASE STUDIES AND RESULTS

Typically, fault diagnosis of IGTs has been based on identifying specific types of faults [27]. In a real industrial setting, there are innumerable things that could potentially go wrong, thus the human cannot be completely removed from the diagnosis loop – indeed, some types of faults are associated to human error and are virtually impossible to predict algorithmically. For example, occasionally the operators shut-off valves accidentally, and this is identified by a safety system as a particular type of mechanical problem [2]. As the response of the operators to the system warnings does not correspond to the actual issue – which is being ignored – a *normal accident* may be generated through a successive chain of inadequate decisions by the operators.

An actual example of a *normal accident* in an IGT is that of debris left inside the fuel ring supplying the combustors, which caused perplexity among the operators. Fuel flow induced the debris to move inside the ring, affecting different combustors at different times. A number of erroneous mechanical causes were considered – all based on previous characterizations of particular faults from sensor readings. An engineer was able to determine the problem only upon dismantling the engine and inspecting the fuel ring.

Therefore, the interest here is on identifying abnormal system behaviour as such, so that potential faulty systems are flagged for further analysis by human experts. In this section, the DREAM (Sec. IV-B) is validated through sensor data from a healthy IGT engine and four faulty IGT cases, exhibiting rotor damage, rotor vibration, compressor damage and worn shaft bearing vibration respectively.

A. Unimpaired IGT Engine

This case study illustrates the behaviour of a healthy IGT engine, and corresponds to independent testing data (Tab. II) spanning a period of approximately 21 days. This data segment

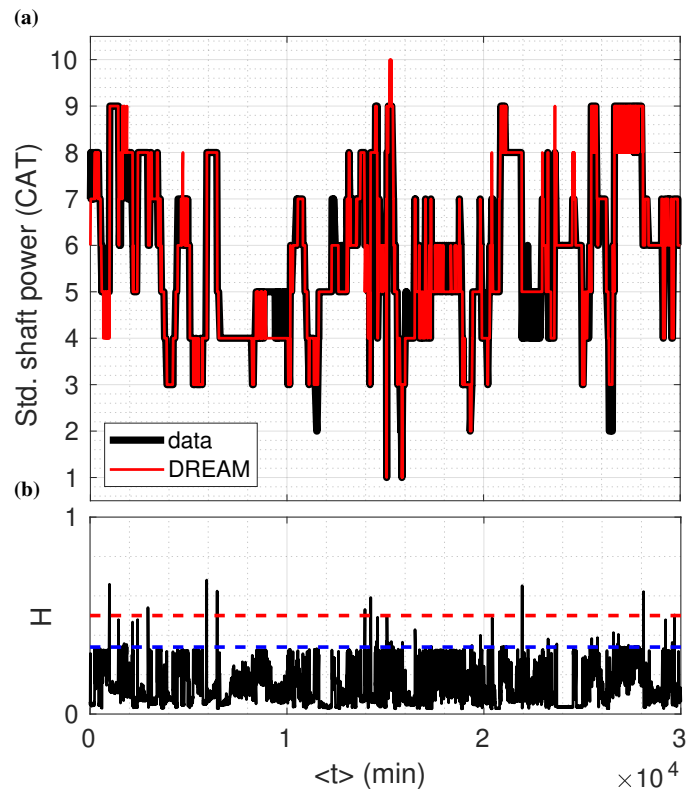


Fig. 6: Standardized shaft power (categorical) produced by a healthy IGT engine during approximately 21 days and output of the DREAM. (b) Entropy of the DREAM prediction ($H^{<t>}$). The dashed lines at $H^{<t>} = 0.34$ (blue) and $H^{<t>} = 0.5$ (red) are empirically established.

was selected because it displays an engine subject to extensive load variations. Nonetheless, the regressor model (Sec. IV-A) is able to accurately predict the power output (Fig. 5).

The same data are analyzed with the DREAM (Sec. IV-B): Fig. 6a shows the prediction for the shaft power as a categorical variable (Tab. III), while Fig. 6b shows the entropy resulting from the softmax layer output (Eq. 8). The *visual* thresholds (dashed lines) in Fig. 6b were empirically set from the currently tested healthy and faulty cases; more than 98% of the samples satisfy $H^{<t>} < 0.34$ for this case study. The exceptions are sporadic spikes due to operator induced variations. Moreover, all the faulty conditions studied next consistently display average $H^{<t>}$ over 0.5. In reality, these threshold values are dependent on the examined system and may also reflect system degradation through time. Hence they should be tuned in an adaptive manner according to the type of system and unit being monitored.

B. Rotor Blade Damage

The blades of the first rotor after the combustors are exposed to the highest temperatures, thus are more likely to suffer damage. The case of an IGT with this particular condition is analyzed in Fig. 7. The DREAM exhibits lower predictability capacity and higher entropy. In the same figure, the time signal segments with higher variability and entropy correspond to

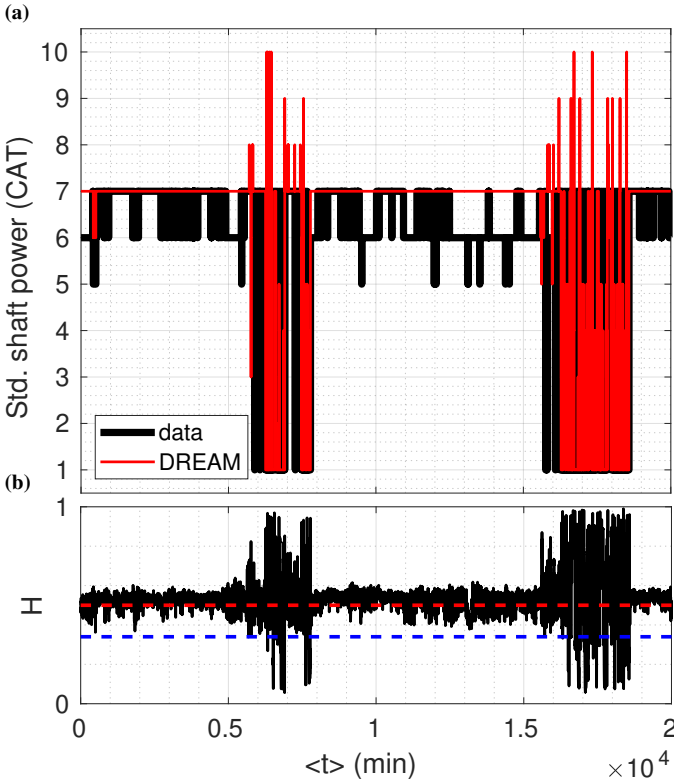


Fig. 7: (a) Standardized shaft power (categorical) produced by an IGT engine with rotor blade damage during approximately 14 days and output of the DREAM. (b) Entropy of the DREAM prediction ($H^{<t>}$). The dashed lines at $H^{<t>} = 0.34$ (blue) and $H^{<t>} = 0.5$ (red) are empirically established.

human induced unpredictability; the operators were trying to restart the machine in order to mitigate vibrations.

C. Rotor Bearing Vibration in Compressor

In another case study, intense rotor vibrations at the compressor turbine were affecting an IGT engine. This was caused by blockage in the lube oil supply to the bearings. Fig. 8 shows the corresponding entropy analysis for this case; while the entropy prediction is larger than for the previous case, the variability is lower. (Fig. 8b), evidencing the reduced predictive capacity of the RNN (Fig. 8a).

D. Compressor Damage

Because compressor discharge air temperature and pressure determine the performance of the combustion in an IGT, compressor efficiency and IGT performance are directly related. The presented case study belongs to an IGT engine suffering compressor damage, possibly as a result of ice formation inside the compressor – through cold air intake. The data analysis for this case is shown in Fig. 9. The DREAM associates large entropy values to this engine after being started, hence the damaged engine is a system with high unpredictability; the model is unsure as to what bin the forecasts should be associated to, because the sensor inputs do not correspond to the typical behaviour in the training set. Interestingly, as a result of reduced efficiency in reality – due to the damage in

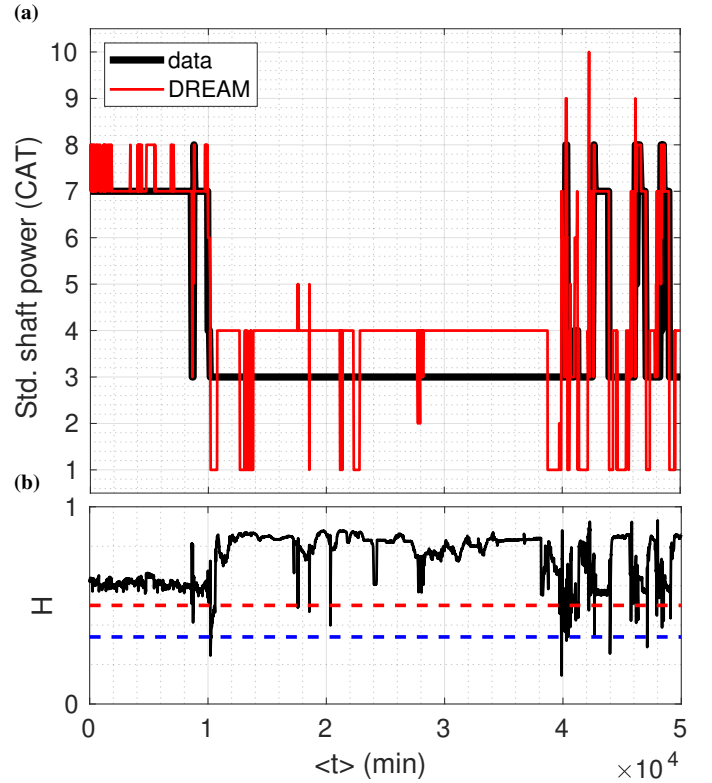


Fig. 8: (a) Standardized shaft power (categorical) of an IGT engine exhibiting large rotor vibration spanning approximately 35 days, and estimated output by the DREAM. (b) Entropy of the DREAM prediction ($H^{<t>}$).

the compressor, the DREAM overestimates the power output. To date, there was no established methodology on detecting compressor icing cases.

E. Worn Bearing Vibration in Power Turbine

Furthermore, the case study of an IGT engine experiencing severe shaft bearing vibration due to lack of lubrication is analyzed. In this case, the actual power output of the engine was not available. Nonetheless, unlike other model-based methods, where the actual output is strictly required in order to calculate the residuals which act as a health indicator [43], the DREAM can predict not only the power output, but also the entropy level with the absence of the real output measures. The DREAM prediction (Fig. 10a) shows when the machine was being shut down and restarted by the operators, to reduce vibration levels. As a result of the stabilization effect of the first restart sequence, the entropy was reduced (Fig. 10b). However, the DREAM indicates relatively low system predictability – or slightly higher entropy – suggesting that there is some system change, e.g. by the replacement of the faulty bearing.

VI. CONCLUSIONS

In nearly all industries, technological development goes hand in hand with an increase in complexity, which is generally associated with a larger number of subsystems, resulting in higher subsystem interactivity. Frequently, considering that the human operators in industrial environments typically interact

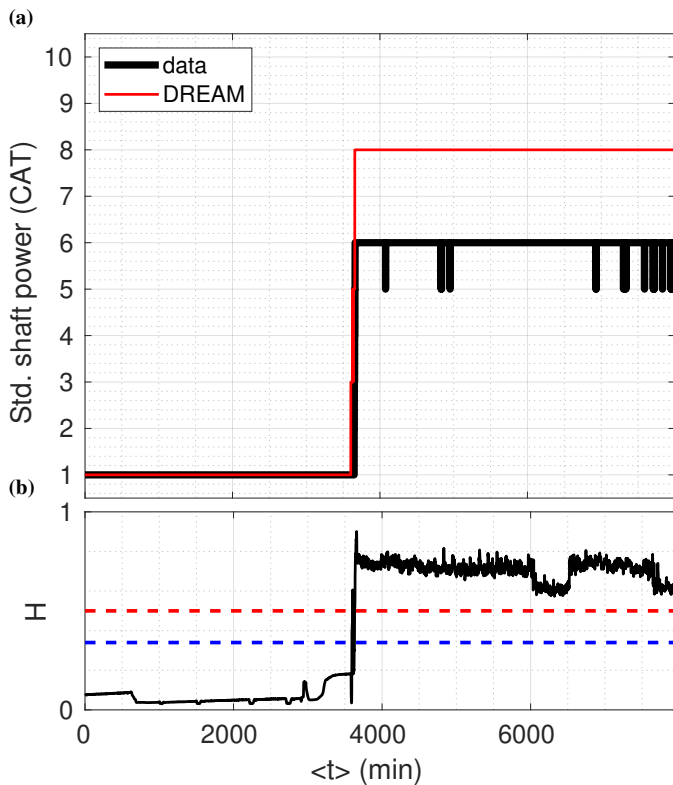


Fig. 9: (a) Standardized shaft power (categorical) produced by an IGT engine with compressor damage during approximately 3 days and output of the DREAM. (b) Entropy of the DREAM prediction ($H^{<t>}$). The dashed lines at $H^{<t>} = 0.34$ (blue) and $H^{<t>} = 0.5$ (red) are empirically established.

with each of the subsystems independently, the nature and degree of the complexity is not obvious or intuitive. In addition, as the relevance of the so-called Industry 4.0 unfolds, cognitive computing techniques are gaining importance, and system complexity is likely to increase even more – and thus, *normal accidents* may become more likely.

With this in mind, a measure to determine the predictability of a complex system, that requires little or no knowledge about the underlying dynamics, is here introduced. The measure consists of using a recurrent neural network to estimate the conditional probability of an output variable, given the system inputs. From the probability function, the information entropy is calculated – indicating how certain is the artificial neural network about the system output, or how predictable is the system. The algorithm is named *Deep Recurrent Entropy Adaptive Model* (DREAM).

The use of an artificial neural network with memory (or recurrent) is justified theoretically – most real-world complex systems present memory and/or hysteresis effects, and experimentally – it is tested that the chosen memory window yields better results than a memory-less model through hyperparameter tuning. For the case of systems that do not present significant memory effects, the recurrent model can be replaced with a non-recurrent neural network architecture – e.g., a convolutional neural network.

The proposed DREAM is validated by testing its efficacy in the particular context of industrial gas turbine engines; it is

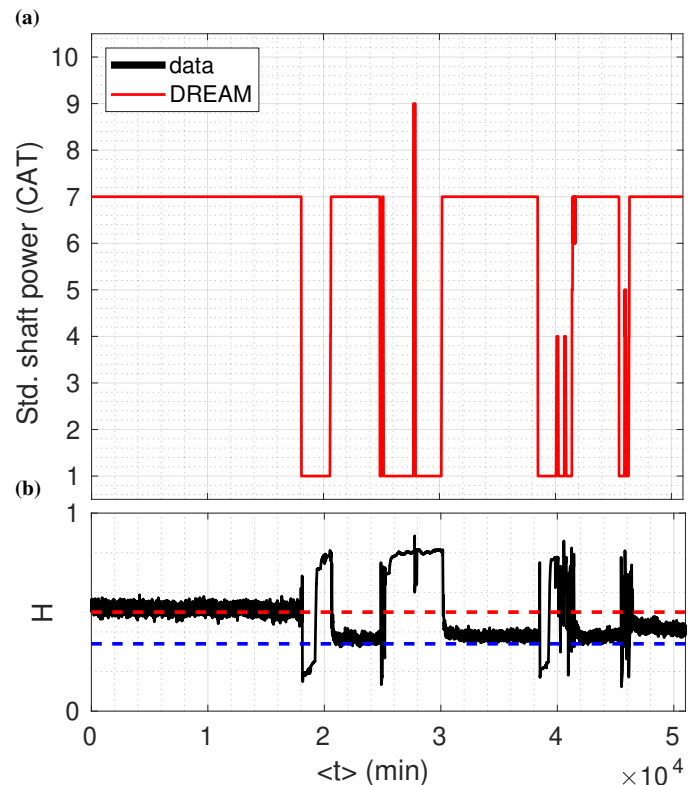


Fig. 10: (a) Standardized shaft power (categorical) of an IGT engine exhibiting large bearing vibration spanning approximately 35 days, and estimated output by the DREAM. (b) Entropy of the DREAM prediction ($H^{<t>}$).

shown that the model is able to discriminate between normal dynamics, corresponding to healthy engines, and anomalous dynamics, corresponding to engines presenting different fault conditions. Moreover, for the case of healthy engines, the given approach can be used as a tool for system efficiency prediction.

The potential applicability of this research covers other types of industries, for example *high risk technologies* such as nuclear, financial technologies (FinTech) and biomedical imaging. Future work is intended towards extending the approach to a broader range of complex systems.

ACKNOWLEDGMENT

This research is supported by the EPSRC Grant EVES (EP/R029741/1). The authors would also like to thank Siemens Industrial Turbomachinery Ltd., Lincoln, U.K., for providing access to the field data.

REFERENCES

- [1] V. Darley, "Emergent phenomena and complexity," *Artificial Life*, vol. 4, pp. 411–416, 1994.
- [2] C. Perrow, *Normal Accidents: Living with High Risk Technologies-Updated Edition*. Princeton university press, 2011.
- [3] C. E. Shannon, "A mathematical theory of communication," *Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.
- [4] A. Ben-Naim, *A Farewell to Entropy: Statistical Thermodynamics Based on Information*. S. World Scientific, 2008.
- [5] M. Zanin, L. Zunino, O. A. Rosso, and D. Papo, "Permutation entropy and its main biomedical and econophysics applications: a review," *Entropy*, vol. 14, no. 8, pp. 1553–1577, 2012.

- [6] Y. Zhang, S. Wang, P. Sun, and P. Phillips, "Pathological brain detection based on wavelet entropy and hu moment invariants," *Bio-medical materials and engineering*, vol. 26, no. s1, pp. S1283–S1290, 2015.
- [7] R. J. Romero-Troncoso, R. Saucedo-Gallaga, E. Cabal-Yepez, A. Garcia-Perez, R. A. Osornio-Rios, R. Alvarez-Salas, H. Miranda-Vidales, and N. Huber, "Fpga-based online detection of multiple combined faults in induction motors through information entropy and fuzzy inference," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 11, pp. 5263–5270, 2011.
- [8] M. Martínez-García and T. Gordon, "Visually interpretable profile extraction with an autoencoder for health monitoring of industrial systems," in *2019 IEEE International Conference on Advanced Robotics and Mechatronics (ARM)*. IEEE, June 2019.
- [9] M. Borowska, "Entropy-based algorithms in the analysis of biomedical signals," *Studies in Logic, Grammar and Rhetoric*, vol. 43, no. 1, pp. 21–32, 2015.
- [10] C. Bandt and B. Pompe, "Permutation entropy: a natural complexity measure for time series," *Physical review letters*, vol. 88, no. 17, p. 174102, 2002.
- [11] E. R. Boer, M. E. Rakauskas, N. J. Ward, and M. A. Goodrich, "Steering entropy revisited," in *Proceedings of the 3rd International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, 2005, pp. 25–32.
- [12] M. Martínez-García and T. Gordon, "A new model of human steering using far-point error perception and multiplicative control," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, Oct 2018, pp. 1245–1250.
- [13] M. Ö. Efe, "Fractional order systems in industrial automation survey," *IEEE Transactions on Industrial Informatics*, vol. 7, no. 4, pp. 582–591, 2011.
- [14] M. Martínez-García, Y. Zhang, and T. Gordon, "Memory pattern identification for feedback tracking control in human-machine systems," *Human factors*, 2019. [Online]. Available: <https://doi.org/10.1177/0018720819881008>
- [15] B. West, M. Bologna, and P. Grigolini, *Physics of fractal operators*. Springer Science & Business Media, 2012.
- [16] K. Suzuki, *Artificial neural networks-Methodological advances and biomedical applications*. BoD—Books on Demand, 2011.
- [17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [18] P. Coulibaly and N. Evora, "Comparison of neural network methods for infilling missing daily weather records," *Journal of hydrology*, vol. 341, no. 1–2, pp. 27–41, 2007.
- [19] R. Lawrence, "Using neural networks to forecast stock market prices," *University of Manitoba*, vol. 333, 1997.
- [20] A. Castano and F. Casacuberta, "A connectionist approach to machine translation," in *Fifth European Conference on Speech Communication and Technology*, 1997.
- [21] M. Catelani, L. Ciani, and M. Venzi, "Rbd model-based approach for reliability assessment in complex systems," *IEEE Systems Journal*, no. 99, pp. 1–9, 2018.
- [22] G. Han, J. Du, C. Lin, H. Wu, and M. Guizani, "An energy-balanced trust cloud migration scheme for underwater acoustic sensor networks," *IEEE Transactions on Wireless Communications*, 2019.
- [23] G. Han, H. Wang, J. A. Ansere, J. Jiang, and Y. Peng, "Sslp: A stratification-based source location privacy scheme in underwater acoustic sensor networks," *IEEE Network*, 2020.
- [24] V. Zaccaria, M. Rahman, I. Aslanidou, and K. Kyprianidis, "A review of information fusion methods for gas turbine diagnostics," *Sustainability*, vol. 11, no. 22, p. 6202, 2019.
- [25] C. Papadimitriou, K. Karamanos, F. Diakonou, V. Constantoudis, and H. Papageorgiou, "Entropy analysis of natural language written texts," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 16, pp. 3260–3266, 2010.
- [26] G. Rosen, "Examining coding structure and redundancy in dna," *IEEE engineering in medicine and biology magazine*, vol. 25, no. 1, pp. 62–68, 2006.
- [27] Y. Zhang, C. Bingham, M. Martínez-García, and D. Cox, "Detection of emerging faults on industrial gas turbines using extended gaussian mixture models," *International Journal of Rotating Machinery*, vol. 2017, 2017.
- [28] R. R. Kline, *The cybernetics moment: Or why we call our age the information age*. JHU Press, 2015.
- [29] M. Xia, T. Li, L. Xu, L. Liu, and C. W. De Silva, "Fault diagnosis for rotating machinery using multiple sensors and convolutional neural networks," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 1, pp. 101–110, 2017.
- [30] M. Xia, T. Li, L. Liu, L. Xu, and C. W. de Silva, "Intelligent fault diagnosis approach with unsupervised feature learning by stacked denoising autoencoder," *IET Science, Measurement & Technology*, vol. 11, no. 6, pp. 687–695, 2017.
- [31] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proceedings of the national academy of sciences*, vol. 81, no. 10, pp. 3088–3092, 1984.
- [32] J. Chung, G. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.
- [33] L. Kaiser and I. Sutskever, "Neural gpus learn algorithms," *arXiv preprint arXiv:1511.08228*, 2015.
- [34] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with lstm recurrent neural networks," *arXiv preprint arXiv:1511.03677*, 2015.
- [35] M. Wielgosz, A. Skoczeń, and M. Mertik, "Using lstm recurrent neural networks for monitoring the lhc superconducting magnets," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 867, pp. 40–50, 2017.
- [36] M. Wielgosz, A. Skoczeń, and E. De Matteis, "Protection of superconducting industrial machinery using rnn-based anomaly detection for implementation in smart sensor," *sensors*, vol. 18, no. 11, p. 3933, 2018.
- [37] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, p. 2451, 2000.
- [38] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [39] J. C. Rodriguez-Quinonez, O. Sergiyenko, F. F. Gonzalez-Navarro, L. Basaca-Preciado, and V. Tyrsa, "Surface recognition improvement in 3d medical laser scanner using levenberg-marquardt method," *Signal Processing*, vol. 93, no. 2, pp. 378–386, 2013.
- [40] J. Rodriguez-Quinonez, O. Sergiyenko, D. Hernandez-Balbuena, M. Rivas-Lopez, W. Flores-Fuentes, and L. Basaca-Preciado, "Improve 3d laser scanner measurements accuracy using a ffbp neural network with widrow-hoff weight/bias learning function," *Opto-Electronics Review*, vol. 22, no. 4, pp. 224–235, 2014.
- [41] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of machine learning research*, vol. 13, no. Feb, pp. 281–305, 2012.
- [42] G. E. Hinton, A. Krizhevsky, I. Sutskever, and N. Srivastva, "System and method for addressing overfitting in a neural network," Aug. 2 2016, uS Patent 9,406,017.
- [43] Y. Zhang, J. Chen, C. Bingham, and M. Mahfouf, "A new adaptive mamdani-type fuzzy modeling strategy for industrial gas turbines," in *Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1599–1603.