Neural Computing and Applications (2023) 35:2991–3009
https://doi.org/10.1007/s00521-022-07167-8

S.I.: APPLICATIONS OF MACHINE LEARNING IN MAINTENANCE ENGINEERING AND MANAGEMENT (IFAC AMEST 2020)

A rare failure detection model for aircraft predictive maintenance using a deep hybrid learning approach

Maren David Dangut[1] · Ian K. Jennions[1] · Steve King[1] · Zakwan Skaf[2]

Received: 29 April 2021 / Accepted: 2 March 2022 / Published online: 26 March 2022

Abstract
The use of aircraft operation logs to develop a data-driven model to predict probable failures that could cause interruption poses many challenges and has yet to be fully explored. Given that aircraft is high-integrity assets, failures are exceedingly rare. Hence, the distribution of relevant log data containing prior signs will be heavily skewed towards the typical (healthy) scenario. Thus, this study presents a novel deep learning technique based on the auto-encoder and bidirectional gated recurrent unit networks to handle extremely rare failure predictions in aircraft predictive maintenance modelling. The auto-encoder is modified and trained to detect rare failures, and the result from the auto-encoder is fed into the convolutional bidirectional gated recurrent unit network to predict the next occurrence of failure. The proposed network architecture with the rescaled focal loss addresses the imbalance problem during model training. The effectiveness of the proposed method is evaluated using real-world test cases of log-based warning and failure messages obtained from the fleet database of aircraft central maintenance system records. The proposed model is compared to other similar deep learning approaches. The results indicated an 18% increase in precision, a 5% increase in recall, and a 10% increase in G-mean values. It also demonstrates reliability in anticipating rare failures within a predetermined, meaningful time frame.

Keywords Predictive maintenance · Deep learning · Extremely rare failure · Auto-encoder · GRU network · Aircraft

1 Introduction

This research is a follow-up to work presented at the 4th IFAC Workshop on Advanced Maintenance Engineering, Services, and Technologies (AMEST 2020) [1]. Unscheduled aircraft maintenance can cause flight cancellation or delay due to the unavailability of spares at the failure location. It can result in unwanted downtime, which increases the airlines' operational costs. Reducing the number of unscheduled maintenance activities through predictive modelling is an excellent initiative for airlines; it reduces maintenance costs and increases fleet availability. According to Airbus [2], by 2025, unscheduled aircraft grounding for fault repairs could cease due to data analytics and operational experience. Aircraft health monitoring and predictive maintenance could enhance the elimination of unscheduled groundings of aircraft by systematically scheduling maintenance intervals more regularly to avoid

✉ Maren David Dangut
maren.dangut@cranfield.ac.uk

Ian K. Jennions
i.jennions@cranfield.ac.uk

Steve King
s.p.king@cranfield.ac.uk

Zakwan Skaf
zskaf@hct.ac.ae

[1] Integrated Vehicle Health Management Centre (IVHM), Cranfield University Bedfordshire, Bedfordshire MK43 0AL, UK

[2] Mechanical Engineering Department, Higher Colleges of Technology, Abu Dhabi, United Arab Emirates

aircraft on ground (AOGs) and the associated operational interruptions [2, 3]. A good predictive model could tell which aircraft parts need schedule checks and those that do not need it, but achieving such maintenance accuracy necessities experience and the right technology [2].

Artificial intelligence (AI) and related technologies, such as the Internet of Things (IoT), machine learning, and symbolic reasoning, have recently advanced to the point where they are causing a paradigm shift in every aspect of human life, including manufacturing, transportation, energy, and advertising. Aerospace is one of the industries that has had the share impact of AI. Aircraft maintenance is quickly adopting AI to build predictive maintenance towards "aircraft smart maintenance". Machine learning algorithms are trained to forecast failure and suggest appropriate actions depending on the predicted failure, which is a step towards smart maintenance solutions. The conditioned-based predictive maintenance provides cost-saving over time-based preventive maintenance Burijs et al. [4] as maintenance is done based on the condition of the component, not time-based as in preventive maintenance. The large amount of data generated from IoT devices installed in aircraft to monitor various components' health conditions combined with data analytics through machine learning can significantly improve aircraft maintenance activities.

Applying correct data analytics and training machine learning algorithms with a vast amount of data can reveal underlying patterns and trends that are not visible to humans. The information discovered can support proactive decision-making, such as recommending the best maintenance actions. Therefore, well-developed machine algorithms are needed to harness relevant information from big data. As artificial intelligence (AI) and related technologies continue to advance, data become more available with a less challenging acquisition, storage, and processing methods. However, newer analytical challenges are emerging. One unique challenge is the extremely rare event prediction when events are infrequent, causing the generated data to be imbalanced, meaning that there are significantly fewer data in one class compared to other classes. Training a traditional machine learning algorithm with a skewed dataset has been shown to degrade the resulting model's performance [4]. Therefore, to develop a robust machine learning model for predictive maintenance, it is vital to address imbalanced data before training (data level approach) or to train the model (algorithm-level approach).

The challenge traditional machine learning algorithms face with the extremely imbalanced dataset is that they are built on the assumption that the data distribution is always balanced, and the cost of misclassification is the same for all classes [5]. However, that assumption is untrue because there exist some domains where the data are highly imbalanced, and the cost of misclassification is high. An example of such a domain is log data generated by the aircraft central maintenance system known as ACMS data. ACMS data are usually imbalanced because aircraft component failure rarely occurs during regular flight operations due to robust safety measures. Apart from the extremely imbalanced problem, ACMS data pose several analytical issues: irregular patterns and trends, class overlapping, and small class disjunct. The standard machine learning algorithm and feature selection or extraction methods become less effective when extremely imbalanced data with class overlapping are used for training [6]. Training machine learning algorithms with imbalanced data has been shown to degrade data-driven models' performance, causing unreliable prognostics [7, 8].

There have been recent improvements in predictive modelling research from both academic and industrial perspectives [9]. There are four types of predictive maintenance modelling approaches, namely physics-based, knowledge-based, data-driven-based, and hybrid-based. The physics-based approach focuses on the equipment degradation process and necessitates the knowledge of the underlying physical failure mechanisms of the components [10]. The physics-based modelling approach's application can be seen in [11, 12], where a digital model of equipment is created to enable the digital-twin (DT) concept in predictive maintenance applications. DT is the concept where multi-physics modelling is combined with data-driven analytics. GE has developed an intelligent IoT-based monitoring and diagnostics platform based on DT to predict physical asset future [13]. The advantage of this approach is that it is applicable even if the dataset is scarce.

Another approach to predictive maintenance modelling is knowledge-based or expert system modelling. This approach involves a combination of domain expert knowledge and computational intelligence techniques. It stores information from domain experts, and rulesets are defined based on the knowledge base for interpretation [14]. The knowledge-based approach has been applied for predictive aircraft maintenance [15, 16]. The authors develop a framework and design methodology for the development of knowledge-based condition monitoring systems. Knowledge-based approaches are more practical for a small and basic system. Its implementation in a big, complicated system, which is difficult and, in some situations, impossible since domain experts must constantly update the rules in the event of upgrades or changes, which is time-consuming.

The data-driven approach involves learning systems' behaviour directly from already collected historical operational data to predict the future of a system's state or identify and match similar patterns in the dataset to infer remaining useful life (RUL) or other insights. The data-

driven modelling methods can be grouped into artificial intelligence (AI)-based, statistical modelling methods, and sequential pattern mining modelling methods [17]. AI methods include machine learning, Bayesian methods, and deep learning methods. AI-based methods have been widely used for developing predictive maintenance models in different industries. Çinar et al. [19] provided a detailed survey on recent applications of AI in predictive maintenance. The hybrid approach includes a combination of two or more techniques for estimation to improve accuracy. Improving accuracy in rare failure prediction requires a robust hybrid approach. In recent times, deep learning (DL) models have been shown to produce state-of-the-art performance when trained with large datasets [18, 19] because of their capability of combining feature extraction with learning. The advances in machine learning research, especially using deep neural networks to learn more complex temporal features, make DL suitable for a large log-based dataset [1]. Other work has shown the effectiveness of DL models in handling extremely imbalanced datasets, especially using log-based ACMS datasets to develop aircraft predictive maintenance models [9].

In this study, a data-driven model is proposed for rare failure prediction. The model consists of deep neural networks, the auto-encoder to detect failures, and bidirectional gated recurrent unit (BGRU) networks combined with convolutional neural networks (CNNs) to learn the co-relationships between variables, enhancing the prediction of rare failure. The effectiveness of the model is evaluated using real-world log-based ACMS time series data. The proposed model will help mitigate the effects of unscheduled aircraft maintenance, producing systematic conditioned-based predictive maintenance, a step towards a smart-aircraft maintenance system.

The remainder of this paper is structured as follows: Section 2 discusses the related work. Section 3 provides a methodology that shows a detailed architecture of the auto-encoder, convolutional neural network—CNN, and bidirectional gated recurrent unit—BGRU. Section 4 presents the experimental set-up and case study. The experimental result is presented and discussed in Sect. 5. Finally, Sect. 6 presents the conclusion and further work.

2 Related work

Deep learning is a branch of machine learning that consists of multiple processing layers that use artificial neural networks (ANNs) to learn data representations at multiple levels of abstraction. Deep learning models have dramatically improved the performance of models in a variety of areas, including large-scale data processing and image identification, among others [7]. The success has been attributed to an increase in the availability of data, hardware, and software improvements and many breakthroughs in algorithm development that speed up training and other data generalisations [20]. Despite the advances, little work has been done to investigate the effect of extremely imbalanced, class overlapping, and small class disjunct on the network's architectures. Many researchers have agreed that the subject of imbalanced data with deep learning is understudied [21–24]. In deep learning, the ANNs are trained to find complex structures in a dataset by using a back-propagation algorithm. The algorithm calculates errors made by the model during training, and the models' weights are updated in proportion to the error. The drawback of this learning method is that examples from both classes are treated the same. In that situation where the data are imbalanced, the model will be adapted more to the majority class than the minority class, which can affect the performance of the models [20]. The majority of the deep learning methods for imbalanced classification have depended on integrating either resampling or cost-sensitive into the deep learning process [25]. For instance, Hensman et al. [26] use random oversampling techniques to balance the data and then train the balanced data using CNN. Also, Lee et al. [22] use random undersampling to balance the dataset for the purpose of pretraining CNN. The use of dynamic sampling to adjust the sampling rate according to the class size for training CNN was proposed by Pouyanfar et al. [27]. Buda et al. [24] investigate the effect of random oversampling, random undersampling, and two-face learning across many imbalanced datasets on deep neural networks. The literature review [20, 24] reveals that most of the proposed deep learning resampling approaches for imbalanced problems use image datasets and CNN architecture. The need to investigate the effect of imbalanced on other deep learning architectures and to use time series is still lacking.

On the other hand, several studies have focused on applying cost-sensitive strategies to solve the problem of imbalanced classification, which entails changing the deep learning process to favour both classes during model training. For example, Khan SH et al. [28] proposed a cost-sensitive deep neural network that can automatically learn robust feature representations for both the majority and minority classes. Also, Zhang et al. [29] propose cost-sensitive deep belief networks, and Wang H et al. [30] propose a cost-sensitive deep learning approach to predict hospital readmission. Also, the use of loss function to control biases has been shown in Wang S et al. [23]. The authors proposed a novel loss function called mean false error and its improved version of mean-squared false error for learning from an imbalanced dataset. Similarly, a new loss function called focal loss was proposed by Lin et al. [31] for dense object detection in image classification. The

focal loss was proposed to specifically handle the challenge of extreme data imbalances commonly faced in object detection problems, where the foreground samples usually outnumber the background samples. Normally, this type of problem is mostly solved using the one-stage detection approach or two-stage detection. The two-stage detection usually performs at the cost of computation time compared to one-stage. Lin et al.'s [31] study focused on determining how the one-stage approach with fast computation time can achieve a state-of-the-art performance compared to the two-stage. Their study discovered that the main cause of performance degradation in one-stage detection is the imbalanced data problem. The overwhelming background samples create imbalance, causing the majority class to account for most of the overall loss. To address that challenge, Lin et al. [8] proposed a loss function known as the focal loss (FL) derived from a normal binary cross-entropy loss. The FL is expressed as follows:

$$\text{Focal Loss FL}\left(p_{,t}\right) = -(1 - (p_t))^{\gamma}\log_{10}(p_t) \quad (1)$$

The new FL tries to reduce the impact that the majority of samples have on the loss by multiplying the cross-entropy loss with a modulating factor $-(1 - (p_t))^{\gamma}$, where the hyperparameter $\gamma \geq 0$ adjusts the learning rate, the negative samples are downweighed. Their implementation shows that using one-stage detection with focal loss by selecting the right learning rate outperformed the two-stage approach. The implantation method was only compared with cross-entropy and tested for imbalance problems in objection detection. The focal loss was later tested in image classification by K Nemoto et al. [32]. The authors use CNN architecture and then compare the performance of focal loss and cross-entropy loss for image classification. The open literature lacks a study investigating the focal loss's effectiveness on time-series systems' log-based datasets, particularly the ACMS dataset.

The identification and prediction of rare failures are active research subjects that have sparked the creation of a variety of methodologies [33]. Asset rare failure prediction is a critical issue that has been approached within various contexts, such as machine learning and statistics [1, 17]. System log data have widely been used to develop rare failure predictive models in different domains. For example, deep learning has been used to predict rare IT software failures using a log-based dataset [34]. Panagiotis et al. [35] developed a failure event model using post-flight records. The authors used multiple instances learning approaches to structure the model as a regression problem to approximate the risk of a target event occurring. Sipos et al. [36] developed a data-driven approach based on multiple-instance learning for predicting equipment failures. Evgeny [10] developed a data-driven rare failure prediction model using event matching for aerospace applications. As seen in the previous study by Maren et al. [1], one of the approaches to identifying and predicting rare failure is using an anomaly detection approach, which is framed in the form of unsupervised machine learning, where the data are divided and labelled as negative and positive samples. In the case of using an auto-encoder, each class is treated separately, and the negatively labelled samples' low-dimensional features are extracted from higher-dimensional data using any feature extraction processes [1]. Then, rare failures are detected and predicted based on the reconstruction error. Most of the well-known traditional or typical data reduction and fault detection methods are the principal component analysis (PCA), partial least square (PLS), and independent component analysis (ICA). These methods use different ways to reduce data dimensionality, and they have achieved a varying degree of success on different data distributions [1, 37]. However, they have fundamental limitations to the nonlinear features since they rely on linear techniques. Kernel tricks have been developed to convert the nonlinear raw data into linear data, and examples are the KPCA [37] and KICA [38]. However, they required high computational power due to kernel function, especially large data [1].

Deep learning (DL) has recently proven superior performance in many areas, such as image classification. Also, it has widely been used in the finance sector for the analysis of time-series data [9]. DL can also be utilised for predictive maintenance. The system installed to monitor an asset's state generates extensive time-series data. Therefore, deep learning algorithms are trained using time-series data to find patterns to predict failures. Recent developments in deep learning have made it easy for deep, complex artificial neural networks to automatically extract features from the original dataset (dimension reduction) during training [39, 40]. The auto-encoder (AE) [41] is an example of a deep neural network algorithm that has been successfully implemented for fault detection and prediction. However, it needs larger data samples and a longer processing time to achieve higher performance [42]. Advances have been made to tackle slightly rare event predictions, especially in the aerospace domain, using machine learning approaches [1, 43, 44]. Deep learning models have also been developed for rare event predictions. For example, Wu et al. [18] developed a weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems. Their model is composed of long recurrent convolutional LSTM model with a sampling policy. Also, Khanh et al. [19] developed a dynamic predictive maintenance framework based on sensor measurements. Changchang et al. [45] combined multiple DL algorithms for aircraft prognostic and health management. In fact, Burnaev et al. [46] pointed out that

many aircraft predictive maintenance solutions are built on basic threshold settings that detect trivial errors on specific components. On the other hand, the threshold-setting strategy is prone to producing high false-positive rates, which lowers model confidence.

Although the approaches mentioned above have successfully handled normal fault detection and prediction, there was a limited study about the application of deep learning models for extremely rare failure prediction, especially for predictive aircraft maintenance using the ACMS dataset. Also, developing a robust predictive model for costly rare aircraft component failure using a large log-based dataset is quite challenging because many components work together and influence each other's lifetime. Another challenge is the heterogeneous nature of the ACMS log data, including symbolic sequence, numeric time series, categorical variables, and unstructured text.

Therefore, our approach focuses on extremely rare failure prediction using log-based aircraft central maintenance system (ACMS) data. Secondly, the work also concentrates on applying a hybrid of deep learning techniques for performance optimisation. The proposed model integrates AE with BGRU and CNN to detect and predict extreme aircraft component replacement. The hybrid method is designed to address the challenge of irregular patterns and trends caused by skewed data distributaries, hence enhancing the prediction of rare failures.

3 Methodology

3.1 Auto-encoder and bidirectional gated recurrent unit network architecture

This section explains how to combine auto-encoder and bidirectional gated recurrent unit network designs to improve predictive model performance using large log-based, multivariate, nonlinear, and time-series datasets.

3.1.1 The auto-encoder (AE)

As presented in Maren et al. [1], auto-encoder [47, 48] is a specific type of multi-layer feedforward neural network where the input is the same as the output neurons. AE aims to learn the original data's internal representation by compressing the input into a lower-dimensional space called latent-space representation (see Fig. 1). It then uses the compressed representation to reconstruct the output while minimising the error for the input data. Training is done using a back-propagation algorithm with respect to the loss function. AE comprises three components: encoder X, latent-space P, and decoder Y. The encoder compresses the input and produces the latent representation. The decoder then reconstructs the input only using the latent representation. An encoder with more than one hidden layer is called a deep auto-encoder.

The encoding and decoding process can be represented using the equation as follows:

$$p_i = f(w_p.x_i + b_t) \tag{2}$$

$$y_i = g(w_y.p_i + b_t) \tag{3}$$

where f(.) and g(.) are the sigmoid functions, w_i represents the weights, and b_i represents biases. The following minimised loss function is used to train the model:

$$L(X, Y) = \frac{1}{2n}\sum_i^n \|x_i - y_i\|^2 \tag{4}$$

where x_i represents the observed value, y_i represents predicted values, and n represents the total number of predicted values.

Equation (3) helps in checking the validity of the resulting underlying feature P.

Figure 1 shows a more detailed visualisation of an auto-encoder architecture. First, the input data pass through the encoder, a fully connected artificial neural network (ANN), to produce the middle code layer. The decoder, which has a mirrored ANN structure, will produce the output using the middle-coded layer. The goal is to get an output identical to the input. Creating many encoder layers and decoder layers will enable the AE to represent more complex input data distribution [1].

3.1.2 The bidirectional gated recurrent unit

A bidirectional gated recurrent unit (BGRU) is a recurrent neural network that has successfully been used to solve time-series sequential data problems because of its bidirectional learning approach, which enhance the learning of temporal patterns in the time-series data [49]. Each BGRU block contains a cell that stores information. Each block is made up of a reset and update gate, and the cells help tackle the vanishing gradient problem Janusz et al. [50]. The reset gate determines how to combine new input with previous memory, while the update gate defines how much of the previous memory to retain, BGRU comprises two GRU blocks. The input data are fed into the two networks, the feedforward and feedback with respect to time, and both of them are connected to one output layer [51]. The gates in bidirectional GRU are designed to store information longer in both forward and backward directions, providing better performance than feedforward networks. The bidirectional approach provides the capability to use both the past and future contexts in a sequence. BGRU can be expressed as:

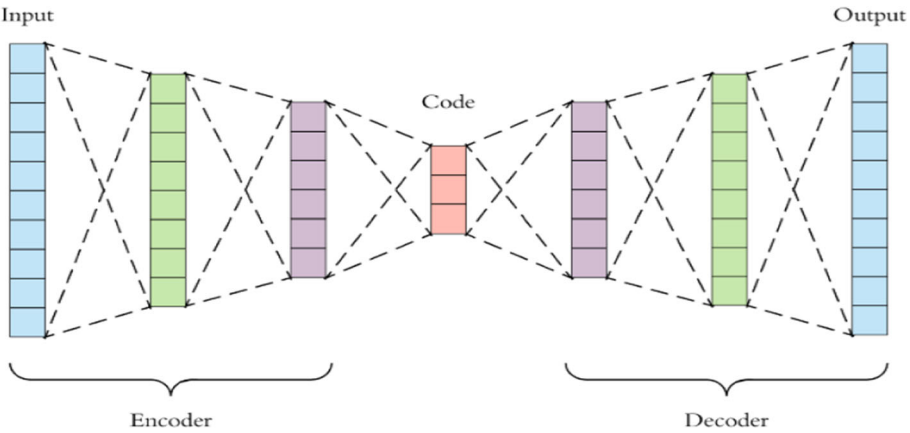


Fig. 1 Auto-encoder architecture [47]

$$h_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{5}$$

where $\overrightarrow{h_t}$, is the feedfoward and $\overleftarrow{h_t}$ the backward block

The final output layer at time t is:

$$y_t = \sigma(W_y h_t + b_y) \tag{6}$$

where σ is the activation function, W_y is the weight, and b_y is the bias vector.

As seen in Figs. 2 and 3, each of the GRU blocks is made up of four components. Input vector x_I with corresponding weights and bias, reset gate r_I with corresponding weight and bias $W_{r,}U_r, b_r$, update gate z_I with corresponding weight and bias $W_{z,}U_z, b_z$ and out vector h_t with its weight and bias $W_{h,}U_h, b_h$. Fully gated unit is represented as follows:

Initially, for t = 0, the output vector is $h_0 = 0$

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \tag{7}$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \tag{8}$$

$$h_t = z_t\, h_{t-1} + (1 - z_t) \otimes \emptyset h(W_h x_t + U_h(r_t \otimes h_{t-1}) + b_h) \tag{9}$$

where $\otimes$ is the Hadamard product. W, U, b are parameter matrices and vectors. σ_g and $\varnothing h$ are the activation functions, σ_g is a sigmoid function, and $\varnothing$ h is a hyperbolic tangent.

The BGRU section of the model is designed as follows. First, the BGRU cells are constructed so that the result of feedforward is computed (F_t) and the feedback propagation (B_t) are merged at the first BGRU layer. Four methods can merge the outcome, concatenation (default), summation, multiplication, and average. In this study, we will compare the performance of each merging method. The merging is represented as follows:

$$O_t^1 = concat(\left(\overrightarrow{F_t}\right), \left(\overleftarrow{B_t}\right)) \tag{10}$$

Such that $\left(\overrightarrow{F_t}\right) = \left(\overrightarrow{h_1}, \overrightarrow{h_2}, \overrightarrow{h_3}, \ldots, \overrightarrow{h_t}\right)$

and $\left(\overleftarrow{B_t}\right) = \left(\overleftarrow{h_t}, \overleftarrow{h_{t+1}}, \overleftarrow{h_{t+2}}, \overleftarrow{h_{t+3}}, \ldots \overleftarrow{h_n}\right)$

Second, a fully connected layer is used to multiply the BGRU network's output with its weight and bias. Then, a Softmax regression layer makes a prediction using input

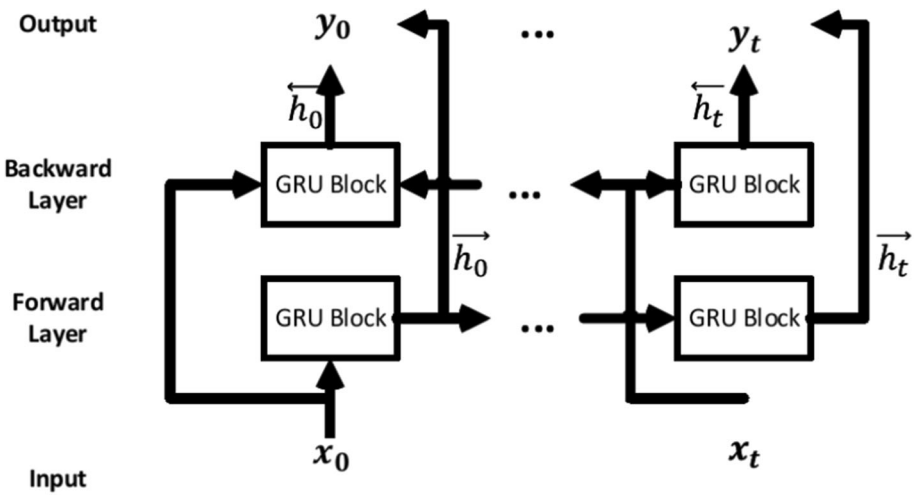


Fig. 2 BGRU architecture with forward and backward GRU layers

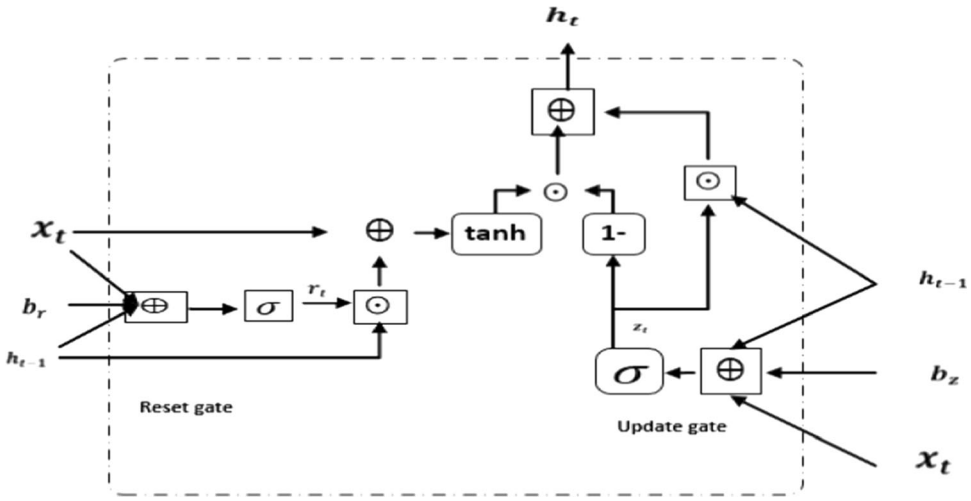


Fig. 3 A GRU block with an update and reset gate, sigmoid and hyperbolic tangent

from the fully connected layer. A weighted classification layer is used to compute the weighted cross-entropy loss function for prediction score and training target, which helps tackle the imbalanced classification problem. The following loss is used:

$$(p_{,t}) = -(1 - (p_t)^{\gamma}) \log_2 (p_t) * \theta_i \tag{11}$$

where $(p_{,t})$ represents the estimated probability of each class, and $\gamma \geq 0$ is the discount factor parameter that can be tuned for best estimation, and θ_i is the logic weight of each class (Table 1).

3.1.3 The convolutional neural networks

The use of deep learning approaches to process time-series data has recently been shown to produce improved results [52]. One of the deep learning approaches that have been widely used is convolutional neural networks (CNNs). CNN's popularity is attributed to its capability to read, process, and extract the most important features of two-dimensional data, contributing to its performance improvement, especially for image classification [53, 54]. In a scenario where the input data are not images, such data can be transformed to suit CNN [55]. Time-series data are one of those data structures that can be transformed for CNN applications. As seen in Fig. 3, with a time-series dataset of length M and width N, the length is the number of timesteps in the data, and the width is the number of variables in a multivariate time series. In transforming the time-series data for CNN [56, 57], a 1D convolutional kernel would be of the same width (number of variables). The kernel will then move top to down performing convolutions until the end of the series. The time-series elements covered at a given time (window) are multiplied with the convolutional kernel elements, the multiplication result is added, and a nonlinear activation function is applied to the value. The resulting value becomes an element of the next new filtered series. The kernel then moves forward to produce the next value. Max-pooling is applied to each of the filtered series of vectors. The vector's largest value is chosen, which is used as an input to a regular, fully connected layer (Fig. 4).

Table 1 Proposed BGRU architecture

Layer (type)	Output shape	Param #
Bidirectional	(Bidirectional multiple	8256
Bidirectional_1	(Bidirection multiple	7872
Repeat_vector	(RepeatVector) multiple	0
Bidirectional_2	(Bidirection multiple	4800
Bidirectional_3	(Bidirection multiple	12,672
time_distributed	(TimeDistri multiple	585
Total params: 34,185		
Trainable params: 34,185		
Non-trainable params: 0		

There is no out-of-the-box or specified rule-of-thumb technique to constructing the framework of BGRU with CNN. Standard artificial neural network structure usually consists of an input layer, one or more hidden layers, and an output layer. The number of hidden layers and neurons used to achieve an optimal solution varies per situation, and it is usually a trial-and-error process. The most common approach is the use of K-fold cross-validation, as seen in [58–60]. However, for evaluation, some k number of nodes need to be defined, which can be obtained by a simple formula,

$$M_k = \frac{M_s}{\alpha(M_i + M_0)} \tag{12}$$

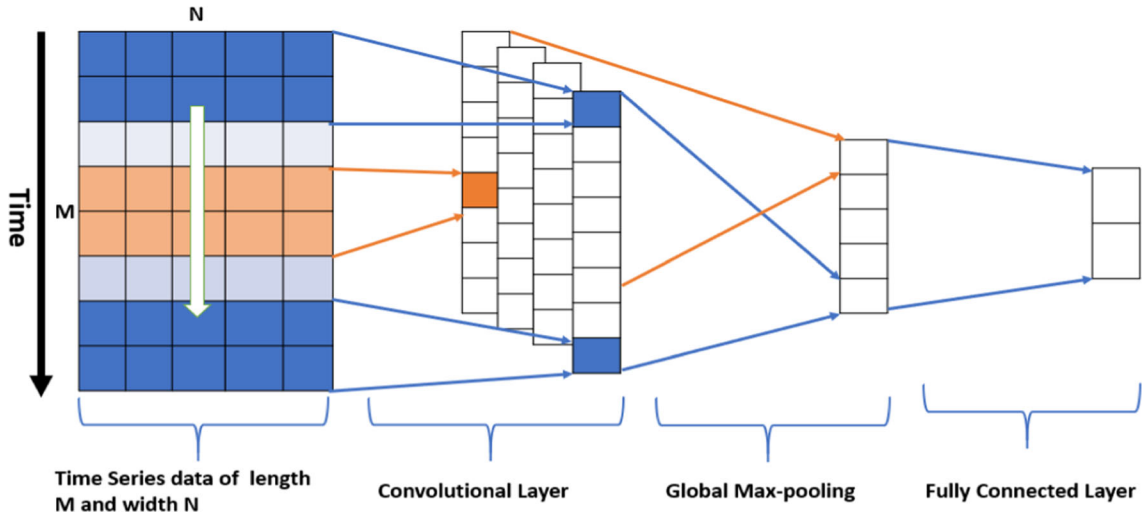


Fig. 4 1D CNN for time-series data

where M_s is the total number of samples in the training data, M_i *and* M_0 are a number of input and output neurons, respectively, and α is the scaling factor. For example, if α is set between two to ten, it means we can calculate eight different numbers to feed into the validation process to obtain an optimal result. The number of parameters to train is computed as Eqs. 5–11, the number of inputs in the first layer equals the defined window size, and the number of folds to use in the cross-validation. The subsequent layers have a number of outputs of the previous layer as input. A simulation is conducted, and the training and testing errors are plotted over the number of neurons in the hidden layer. The number of neurons is chosen that minimises the test error while keeping an eye on overfitting. Because the problem is formulated as binary classification and the data are extremely imbalanced, we use a modified loss function (Eqs. 6–10) and Softmax as the final activation function.

3.2 Proposed method

3.2.1 AE–-CNN–BGRU network

Our goal is to create a model that can identify and predict rare failures using a large log-based dataset. The main idea is to separate the prediction of rare failure from its detection, as shown in Fig. 5. As a result, the proposed model uses two stages: auto-encoder for detecting unusual failures and BGRU and CNN architectures for forecasting future instances of that failure.

The BGRU was chosen in the design because it can capture a long dependency in both directions (forward and backwards) to allow for successful learning. The rationale for the method selection is based on the dataset's characteristics (i.e., heterogeneous and time series in nature). Recurrent neural networks (RNNs) are commonly used to train time-series datasets; nevertheless, RNNs suffer from vanishing gradient difficulties and have a short-term memory. When using a gradient-based learning strategy with back-propagation to train a deep multi-layer RNN (feedforward network), the problem of vanishing gradients emerges [61]. Each iteration of the method updates the weight of each ANN in proportion to the partial derivatives of the error function with respect to weight. The problem develops when valuable gradient information cannot be propagated back to the model's input layer from the out-layer [62]. The gated recurrent unit (GRU) networks were designed to capture long-time dependencies in sequence learning and to manage the gradient vanishing problem using modified hidden layers or gates in order to overcome the vanishing gradient problem in RNN.

Convolutional neural network (CNN) uses a process known as convolution when determining a relationship between available variables in the dataset [20]. For example, in convolutional learning, given two functions f and g, the convolution integral expresses how the shape of one function is modified by the other. Traditionally, CNNs were designed to process multi-dimensional data, such as image classification, not to account for sequential dependencies like in RNNs, LSTMs, or GRUs [63]. Therefore, the key benefit of adding CNN layers for sequential learning is its ability to use filters bank [64] to compute dilations between each cell, also referred to as "dilated convolution", which in turn allows the network layers in CNN to understand better the relationships between the different variables in the dataset, generating improved results.

As explained in Maren et al. [1], the dataset is extremely imbalanced; that is, the imbalanced ratio between the positively labelled and negatively labelled data is less than

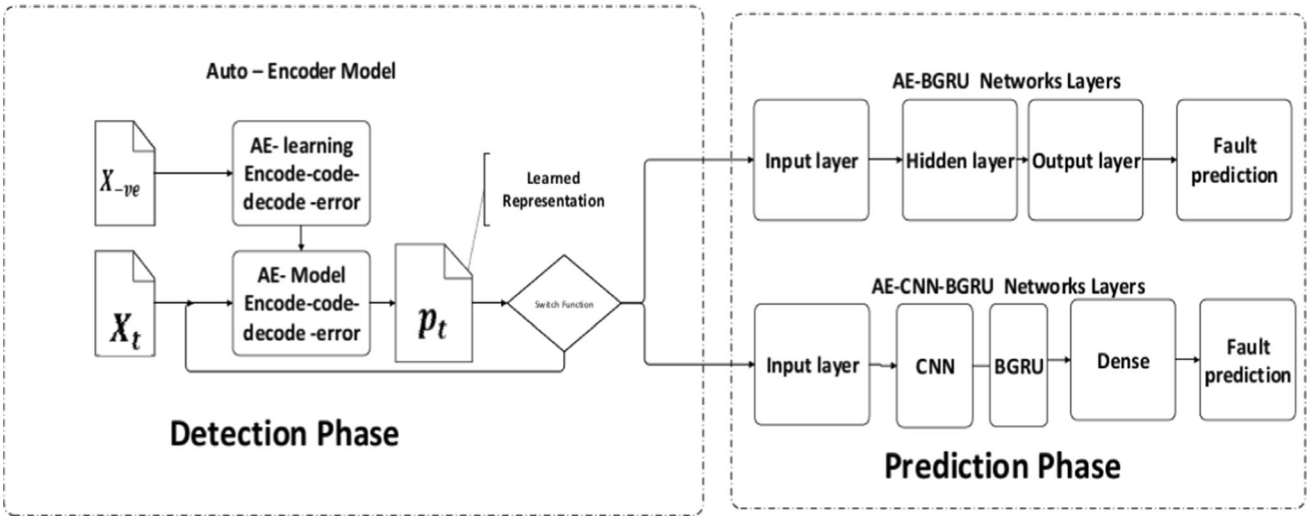


Fig. 5 An integrated AE, BGRU, and CNN networks for rare fault detection and prediction

5% of the total. In such an extremely rare problem, traditional deep learning algorithms are overwhelmed by the majority class, producing bias result without detriment to the minority class [41, 65]. Therefore, we proposed AE–CNN–BGRU to handle the problem differently. The framework of the proposed model is shown in Fig. 5. The first AE model is used to detect rare failures using reconstruction errors at the detection stage. The data are divided into positively labelled (rare minority class) and negatively labelled (majority class). The AE model is then trained with only negatively labelled data (X_{-ve}) by feeding the encoder layer of AE with the original negatively labelled data. The latent code, which represents a compresses feature, is extracted in the middle layer. The decoder layers will then reconstruct the original data using compressed latent code as input. After the encode–decode process, a reconstruction error is known, which also shows the highest error that is later used for threshold setting. Since the AE model is first trained using negatively labelled data when the data are combined (X_t) and fed into the AE model. An anomaly can easily be detected because any data point coming from the negatively labelled class is expected to have a low error, and if coming from a positive class, the error will be higher. The low error is because it is coming from the same data used to train the first-section AE model (as seen in the detection phase of Fig. 5). On the other hand, when a new data point is from a positively labelled class, it is expected to have a higher reconstruction error score which will pick as an anomaly [1].

For example, when a datapoint $\boldsymbol{x_t}$ is fed into the AE model, it will be classified as a fault if the reconstruction error exceeds a defined threshold; otherwise, it will be classified as no-fault. Once the faults are identified, the resulting compressed data are then fed into the next section of the framework, which is the AE–BGRU or AE–CNN–BGRU model for the failure prediction. The input data to the prediction model are the learned latent representation of the original dataset. To determine a threshold that offers the best result, we construct a function that iterates through a loop using precision and recall until the desired threshold is obtained.

4 Case study and experimental setup

The goal of the experiment is to see how well our proposed technique handles the infrequent incidences of failure. The primary research question is whether AE–CNN–BGRU can beat traditional unidirectional deep learning time-series approaches with explicit failure detection and additional training capabilities on an extremely imbalanced dataset. Another significant question is whether learning in two directions might increase model performance for rare failure prediction (feedforward and feedback propagation). Also, how different does the architecture of deep learning models treat the input data? A series of experiments are conducted to answer the questions. A log-based data from the aircraft central maintenance system, which includes aircraft failure and warning alerts, are used. The following experiments are set up.

1. To investigate whether the proposed AE–BGRU model has a performance advantage over the normal GRU model in predicting rare aircraft component failure.
2. To investigate whether additional layers of training in the AE–CNN–BGRU model architecture can improve model performance.
3. To investigate whether training the proposed model using an extremely imbalanced dataset in a

bidirectional way (forward and backwards) can improve model performance.

4. To provide a performance analysis of deep learning architecture for the rare failure prediction via the log-based ACMS dataset.

The modelling approach is divided into two categories: binary class and multi-class. We characterised the first situation as a multi-class classification problem that predicts all the targeted component failures at once. Second, we modelled it as a binary classification problem in which specific functional items are predicted.

4.1 Dataset

As Maren et al. [66] explained, this study uses over eight years' worth of data recorded from more than 60 aircrafts. The dataset is collected from two databases. The first database is the aircraft central maintenance system (ACMS) data, which comprises error messages from BIT (built-in test) equipment (that is, aircraft fault report records) and the flight deck effects (FDE). These messages are generated at different stages of flight phases (take-off, cruise, and landing). The second database is the record of aircraft maintenance activities (i.e., the comprehensive description of all aircraft maintenance activities recorded over time). The dataset is obtained from a fleet comprised of A330 and A320 aircraft. Some components are identified by functional item number (FIN) chosen for validation. The target components are chosen based on their high practical value and an adequate number of known failure cases. The other consideration for the choice of the component is those that are replaced due to unscheduled. Figure 6 shows an example of the ACMS dataset.

Data from the year 2011 to 2016 are used for training, while the remaining data from 2016 to 2018 are used for testing. The targeted LRUs from the A330 aircraft family are **4000KS**—Electronic Control Unit/ Electronic Engine Unit, **4000HA**—Pressure Bleed Valve, and **438HC**—Trim Air Valve. From A320 are **11HB**—flow control valve, **10HQ**—Avionics equipment ventilation computer, **1TX1**—air traffic service unit.

4.2 Evaluation metrics

In general, "accuracy" is the most important performance metric in machine classification. However, using accuracy to measure performance in extreme imbalanced classification issues can be misleading since, in order to attain high overall accuracy, classifiers would be biased towards the majority class. As a result, various alternative metrics, like precision, recall, g-mean, and area under the curve, are used better to evaluate the classifiers' performance [67].

From Fig. 7, we derive Eqs. (12) to (16),

$$\text{Accuracy} = (\text{tp} + \text{tn})/n \tag{12}$$

$\Rightarrow$ ability to correctly classify all observations.

Precision (p): is the measure of classifier exactness, the percentage of true-positive predictions made by the classifier that is truly correct. So, low precision indicates a large number of false positives.

$$P = tp/(tp + fp) \tag{13}$$

Recall (r) is the classifier completeness measure and is defined as the percentage of true positives that the classifier can correctly detect. So, low recall indicates many false negatives.

$$\text{R} = tp/(tp + fn) \tag{14}$$

G-mean measures the root of the product of class-wise sensitivity; it maximises each class's accuracy and keeps the accuracy balanced.

$$G - \text{mean} = \sqrt{P * R} \tag{15}$$

False-positive rate is calculated using the following equation.

$$\text{FPR} = \text{fp}/\text{fp} + \text{tn} \tag{16}$$

Receiver operating characteristic curve (ROC): ROC is a graphical representation that illustrates the classifier's diagnostic ability as the discriminant threshold is varied. An excellent model has an area under the curve AUC with a value near one, meaning the model has a good separability measure.

Assuming we have two classes, the positive and negative classes ROC curve of those classes' probability is discribed in Figs. 8 and 9.

Figure 8 shows the ROC curve for an ideal situation. The green distribution curve represents the positive class (component failure), and the black distribution curve represents the negative class (non-failure). When the two curves do not overlap, the model has an ideal measure of separability (that is, the model can correctly distinguish between positive and negative classes).

Figure 9 depicts a case in which two distributions intersect. Type 1 and type 2 faults will be introduced in this instance. The mistake can be minimised or maximised depending on the threshold value. When the AUC is 0.8, the model has an 80% chance of correctly distinguishing between positive and negative data. The weakest separability measure is when the model AUC = 0. (The model is reciprocating the classes, which means the model predicts a negative class as a positive class and vice versa.) When AUC = 0.5, the model has no ability to distinguish between classes.

Fig. 6 Example of the real ACMS dataset. Sensitive data elements have been masked

Event Date	Source	Tail_Number	Failure Message	Message Type
19/12/2013 13:03	Component 1		FDIU	Information
19/12/2013 13:03	Component 2	S1	NO CIDS 2 DATA (INTM)	Warning
20/12/2013 00:59	Component 1		NO CIDS 2 DATA (INTM)	Warning
20/12/2013 06:12	Component 1		NO CIDS 2 DATA (INTM)	Fail
20/12/2013 11:17	Component 3		HPTC VLV(POS)	Information
20/12/2013 11:17	Component 2	S2	ENG 1 FADECXX	Warning
20/12/2013 11:17	Component 2		NO CIDS 2 DATA	Warning
20/12/2013 15:29	Component 1		RADAR1 ANTENNA	Fail
20/12/2013 15:29	Component 1		RADAR1 CONTROL UNIT	Information
20/12/2013 15:29	Component 1		RADAR1 TRANSCEIVER	Warning
20/12/2013 22:31	Component 2	S3	ENG REV SETXX	Warning
21/12/2013 06:10	Component 1		FUEL R TK PUMP 1+2 LO PR	Fail
21/12/2013 06:10	Component 1		AFS:ELAC2	Information
21/12/2013 07:06	Component 3		AFS:MCDU2	Warning
21/12/2013 07:06	Component 2		AFS:MCDU2(FW DISC)/FMGC1	Warning
21/12/2013 07:06	Component 2		AFS:MCDU2(FW DISC)/FMGC2	Fail
21/12/2013 19:49	Component 1	S2	FWC1 :NO DATA FROM ECU2A	Information
21/12/2013 19:49	Component 1		NAV ALTI DISCREPANCYXX	Warning
21/12/2013 23:12	Component 1		AUTO FLT AP OFFX	Warning
21/12/2013 23:12	Component 2	S3	AFS:FMGC2	worning
22/12/2013 05:11	Component 1		FUEL L TK PUMP 1 LO PR	Information
22/12/2013 05:11	Component 1		AUTO FLT AP OFFX	Warning
22/12/2013 05:11	Component 3	S1	FDIU	Warning
22/12/2013 22:42	Component 2		NO CIDS 2 DATA (INTM)	Fail
22/12/2013 22:42	Component 2	S2	NO CIDS 2 DATA (INTM)	Information

Fig. 7 Confusion matrix and ROC curve

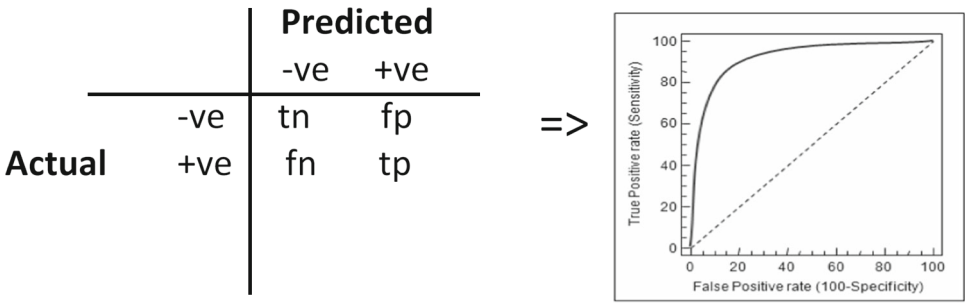
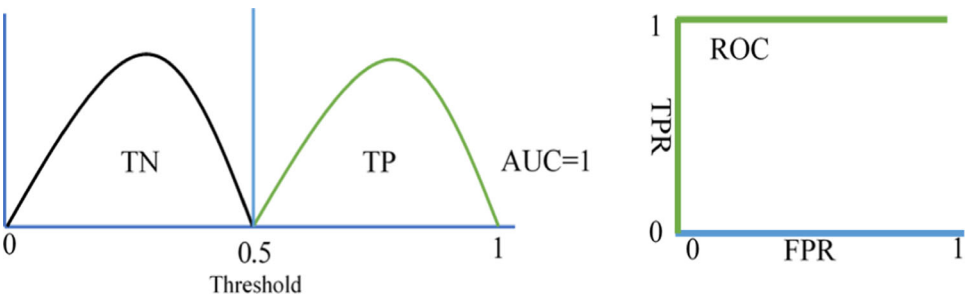


Fig. 8 ROC curve for an ideal situation



4.3 Sensitivity analysis for BGRU merge modes

Sensitivity analysis was carried out to determine the best merging mode that can be used to integrate the outcomes of the BGRU layers for the proposed model. As shown in Fig. 10, plotting loss against epoch, the line plot is created to compare the four merge modes (summation, concatenation, multiplication, and average). A time-series data of size 10,000 was generated and trained, using a loss shown in Eqs. (5–10) and running the BGRU networks for 200 epochs. The result indicates that concatenation (the green line) is the best merge mode because it has lower loss values.

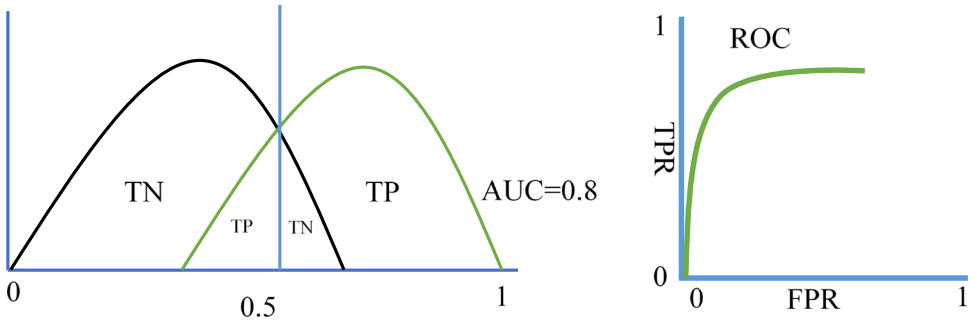


Fig. 9 ROC curves showing overlap distributions with AUC = 0.8

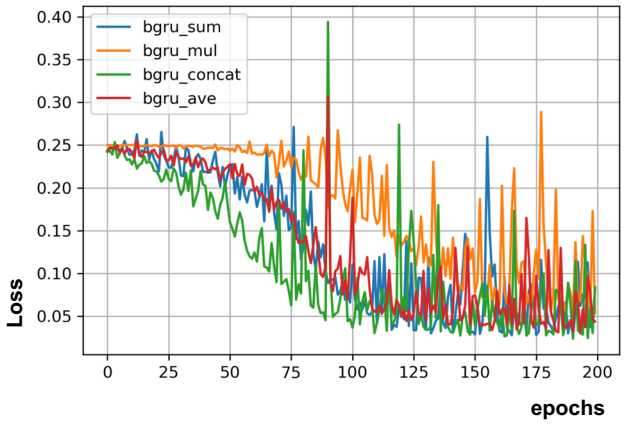


Fig. 10 Comparing BGRU merge modes. The figure shows the analysis to determine the merging mode that can be used for the BGRU layers in the proposed AE–CNN–BGRU model. The target is to choose the best merging method (i.e., with lower error)

Furthermore, an analysis was carried out to determine the effect of bidirectional networks as compared to unidirectional ones. Three network architectures were set up for the analysis: two unidirectional (the forward and the backwards networks) and the bidirectional network. The result is shown in Fig. 11; as observed, the GRU forward and GRU backward show a similar pattern, while BGRU_concat (green) shows a better loss (low errors). The

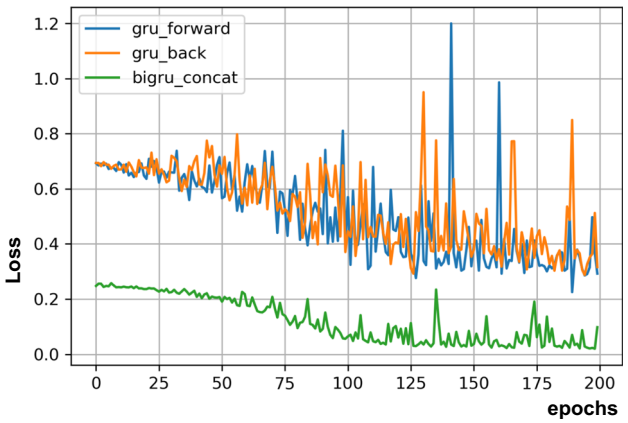


Fig. 11 Comparing GRU with BGRU

comparison result indicates that BGRU can add performance improvement, not just merely reversing the input sequence.

5 Result and discussion

A study is conducted to determine whether training the model using an imbalanced dataset bidirectionally with focal loss can improve the minority class's detection. Two bidirectional models were considered, the AE–BGRU and the AE–CNN–BGRU models, and compared with GRU (baseline); the result is shown in Table 2. The models are validated using data from two aircraft families, the A330 and the A320 groups; the size of the training dataset is 360575 and 389,829, respectively. The target is to predict the replacement of aircraft LRU identified by their functional identification numbers (FINs). The validation result is based on the validation (testing) data, and the size is dependent on the number of patterns related to each target component. The targeted number of failures for each component are 4000KS = 11, 4000HA = 13, 438HC = 9, 11HB = 6, 10HQ = 8 and 1TX1 = 15.

As observed in Table 2, the proposed models show superior performance compared to the baseline. Considering the A330 dataset and training the proposed algorithms to predict each component's failure, it can be observed after validation. The result for predicting failure of 4000KS (the aircraft electronic engine unit) using the AE–BGRU model records a precision of 72%, recall of 61%, g-mean 67%, and a false-positive rate of 0.091%. AE–CNN–BGRU model achieves a precision of 90%, recall of 66%, g-mean of 77%, and a false-positive rate of 0.011%. Compared to normal GRU with a precision of 60%, recall 0.55%, g-mean 53%, and a false-positive rate of 0.005, a similar result is seen for the other components, the 4000HA (pressure bleed valve) and the 438HC (trim air valve).

Using data from the A320 aircraft family: the results also indicate superior performance for the proposed AE–BGRU and AE–CNN-BGRU models as compared to unidirectional GRU. The result for predicting the failure of 11HB (flow control valve) indicates that AE–CNN–BGRU

Table 2 Aircraft A330 and A320 rare failure prediction of individual LRUs using ACMS dataset

Aircraft ACMS dataset														
	LRU's	IR	GRU (Baseline)				AE–BGRU				AE–CNN–BGRU			
			P	R	GM	FPR	P	R	GM	FNR	P	R	GM	FNR
A330-Family	4000KS	0.0043	0.60	0.55	0.53	0.005	0.720	0.61	0.67	0.00091	0.909	0.66	0.778	0.00011
	4000HA	0.0047	0.41	0.40	0.41	0.008	0.538	0.538	0.632	0.00127	0.769	0.768	0.769	0.000638
	438HC	0.0044	0.54	0.51	0.53	0.006	0.666	0.600	0.632	0.00083	0.88	0.610	0.730	0.00027
A320 Family	11HB	0.0028	0.62	0.51	0.49	0.005	0.660	0.58	0.624	0.00019	0.66	0.59	0.671	0.00019
	10HQ	0.0031	0.60	0.51	0.55	0.006	0.625	0.49	0.55	0.00028	0.75	0.66	0.707	0.000191
	1TX1	0.0064	0.66	0.52	0.58	0.007	0.866	0.764	0.814	0.00029	0.85	0.741	0.860	0.000193

****LRUs** represents an aircraft line replacement unit. *P* is precision, *R* is recall, *GM* is g-mean, *FPR* is a false-positive rate

achieved a precision of 66%, recall 59%, g-mean 67%, and a false-positive rate of 0.019% compared to GRU with a precision of 61%, recall 51% g-mean 49%, and a false-positive rate of 0.005. Similar performance is seen for other components, the 10HQ—Avionics equipment ventilation computer and 1TX1—Air traffic service unit.

In the six FINs considered, the proposed models show a significant improvement in reducing the false-positive rate, which is very important for any predictive maintenance model acceptability. Also, the AE–CNN–BGRU model shows an overall improvement of 18% in precision, 5% in the recall, and 10% in G-mean.

5.1 Measuring the success rate of the proposed models using A330 aircraft

Figure 12 shows the ROC curve for the proposed models AE–BGRU and the AE–CNN–BGRU. The ROC curve for the AE–CNN–BGRU model (Fig. 12b) shows that AUC = 0.822 indicates that there is an 82.2% chance that the model will be able to distinguish between positive class (component failure) and negative class (non-failure). In contrast, Fig. 12a shows the ROC curve for the AE–BGRU model with AUC = 0.737, which indicates that the model has a 73.7% chance of distinguishing between classes.

Also, to measure the model success rate in predicting extremely rare failure, a confusion matrix was plotted for both proposed models. Figure 13 shows a confusion matrix for predicting the failure of the electronic engine unit (FIN_4000KS). As seen in Fig. 13a AE–BGRU model predicted eight failures correctly out of the eleven true failures, and Fig. 13b shows that the AE–CNN–BGRU model predicted ten out of eleven. This prediction includes 10 flight legs in advance. It can also be observed that the AE–CNN–BGRU model predicts approximately 94% of extremely rare failure of components, which is a reasonable

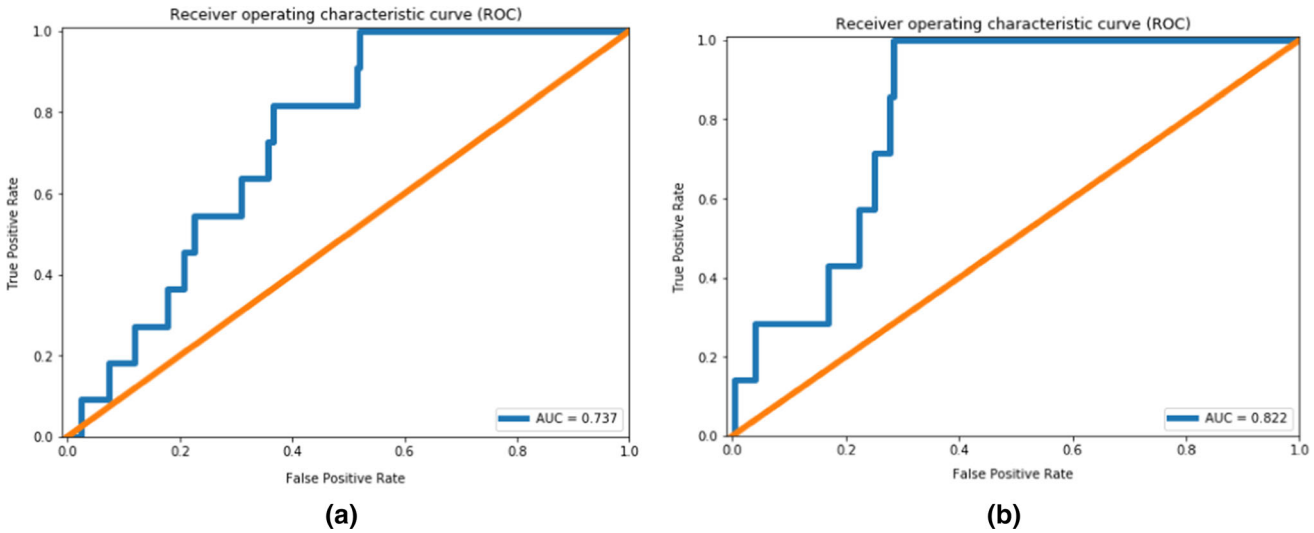


Fig. 12 ROC curve for FIN_4000KS prediction using (**a**) AE–BGRU and (**b**) AE–CNN–BGRU models

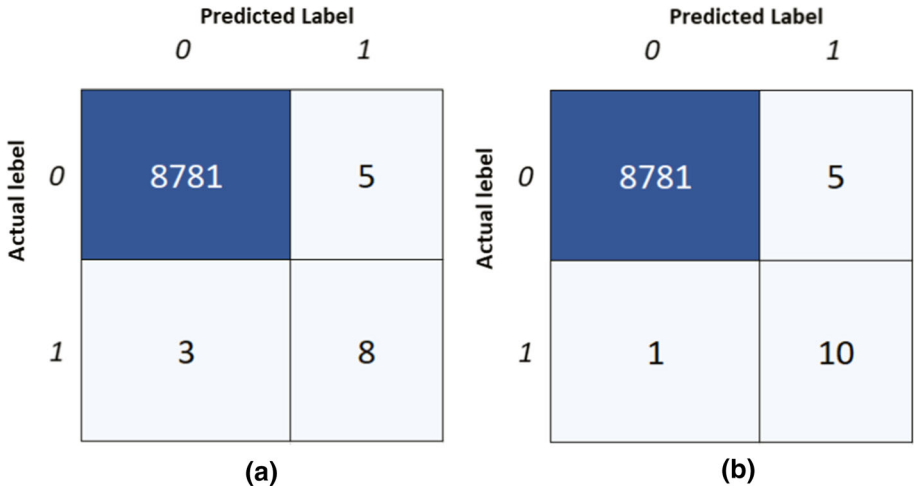


Fig. 13 Confusion matrix for FIN_4000KS using (**a**) AE–BGRU and (**b**) AE–CNN–BGRU model

specificity, especially for aircraft maintenance acceptability.

Similarly, as seen in Fig. 14a, AE–BGRU predicted 7 out of 13, and in Fig. 14b AE–CNN–BGRU predicted 10 out of 13 unplanned replacement of pressure bleed valve (FIN_4000HA) failures. This prediction includes 10 flight legs in advance, and it can also be observed that the AE–CNN–BGRU model shows superior performance. A similar performance is observed for other components tested. The general result indicated that the proposed AE–CNN–BGRU model detected and predicted approximately 80% of extremely rare failures, which is a reasonable specificity, especially for aircraft maintenance.

5.2 Measuring the success rate of the proposed models using A320 aircraft

Figure 15 shows the ROC curve for the proposed models AE–BGRU and the AE–CNN–BGRU. The ROC curve for the AE–CNN–BGRU model (Fig. 15b) shows AUC = 0.864, which indicates that there is an 86.4% probability that the model will be able to distinguish between positive classes (component failure) and negative class (non-failure). In contrast, Fig. 15a shows the ROC curve for the AE–BGRU model with AUC = 0.817, which indicates that the model has an 81.7% probability of distinguishing between classes. The result indicated that AE–CNN–BGRU has an 8% better classification performance compared to AE–BGRU.

Similarly, as seen in Fig. 16a, AE–BGRU predicted 4 out of 6 and Fig. 16b AE-CNN-BGRU 4 out of 6 unplanned replacement of pressure bleed valve (FIN_11HB). This prediction includes 10 flight legs in advance. A similar performance is observed for other components tested. The general result indicated that the proposed AE-CNN-BGRU model detected and predicts approximately 50% of extremely rare failures.

Although both models predicted 50% of the failure, it can be observed that the AE–CNN–BGRU model shows superior performance in terms of recall. A good recall

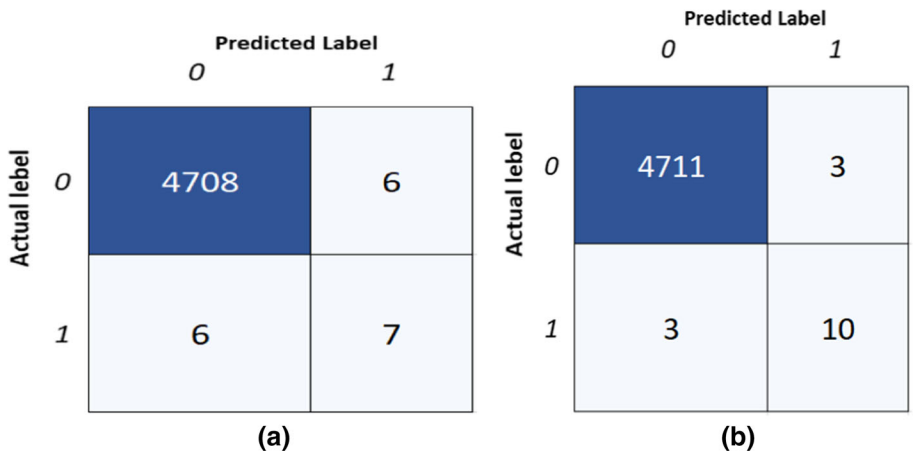


Fig. 14 Confusion matrix for FIN_4000HA using AE–BGRU and AE–CNN–BGRU model

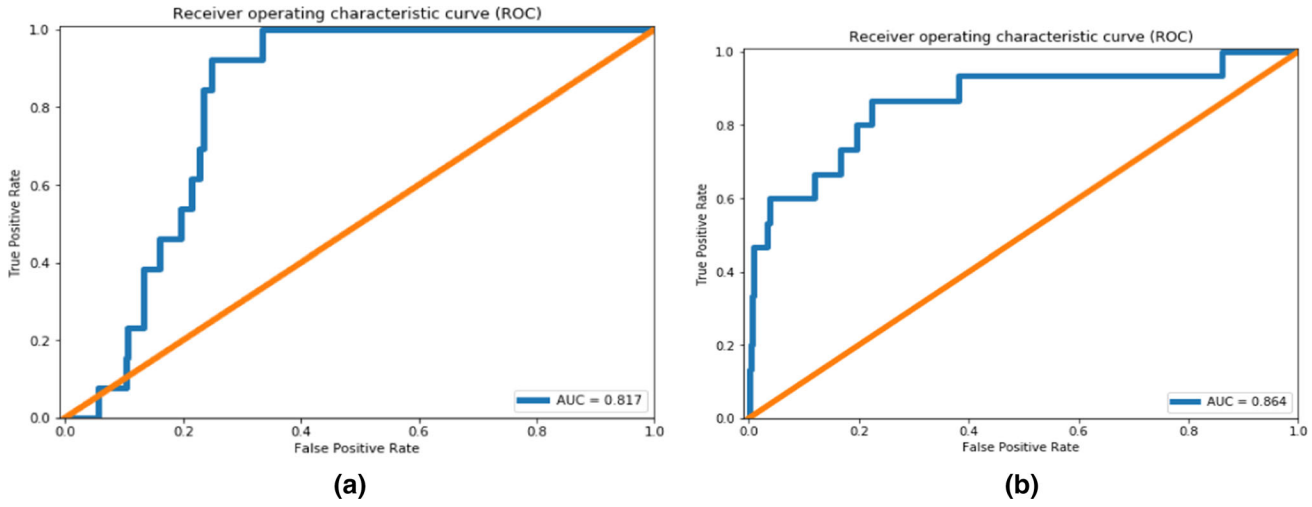


Fig. 15 ROC curve for predicting 11HB using AE–BGRU

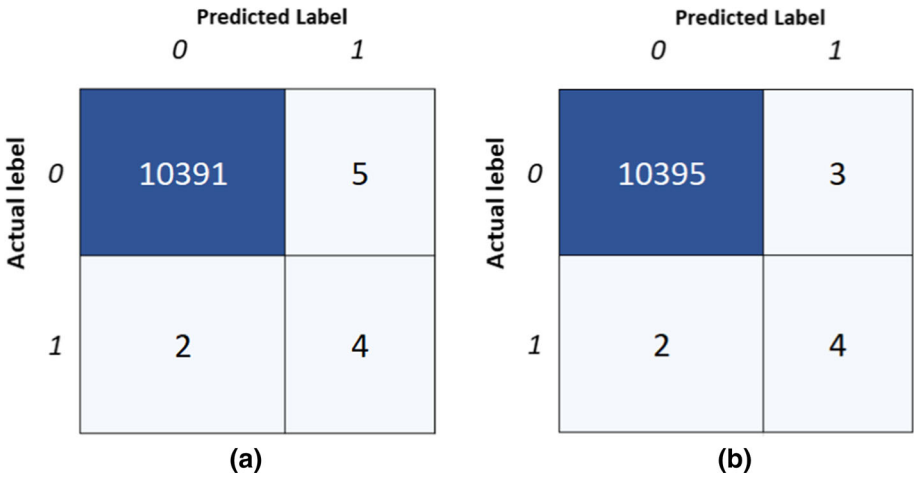


Fig. 16 Confusion matrix for FIN_11HB using AE–BGRU and AE–CNN–BGRU model

indicates that the model has a good potential measure of correctly identifying true positives.

5.3 Sensitivity of AE–CNN–BGRU model to design parameters

Additional analysis was carried out to determine whether adding CNN layers to the AE–BGRU network could improve performance. After the implantation, the result indicated that there was performance improvement. The AE–CNN–BGRU model performance improvement can be accounted to the following factors. First, in training time-series dataset, especially using BGRU or LSTM, such networks account for the sequential dependency in a situation where a correlation exists between the variables in the given dataset (a process known as autocorrelation); during training, a normal GRU/LSTM network would treat all the variables as independent, excluding any relationship that exists between both observed and latent variables, whereas CNN uses a process known as convolution when determining a relationship between available variables in the dataset [20]. For example, in convolutional learning, given two functions f and g, the convolution integral expresses how the shape of one function is modified by the other. Traditionally, CNNs were designed to process multi-dimensional data, such as in image classification, not to account for sequential dependencies like in RNNs, LSTMs, or GRUs [63]. Therefore, the key benefit of adding CNN layers for sequential learning is its ability to use filters bank [64] to compute dilations between each cell, also referred to as "dilated convolution," which in turn allows the network layers in CNN to understand better the relationships between the different variables in the dataset, generating improved results.

5.4 Sensitivity analysis of imbalanced ratio.

A sensitivity analysis was carried out for the imbalanced ratio on the designed network architecture and the input data. As observed in Table 2, the six cases considered have different imbalanced ratios (400KS = 0.0043, 4000HA = 0.0047, 438HC = 0.0044, 11HB = 0.0028, 10HQ = 0.0031, 1TX1 = 0.0064). The components differed not only in the imbalanced ratio but also in distributions and failure patterns. As seen in Fig. 17, it can be observed that the novel model (AE–CNN–GRBU) shows a significant reduction in the false-negative rate as compared to others, indicating that it is robust to different conditions of the dataset. Also, it is observed that the imbalance ratio impacts the false-negative rate for the test components from the A330 aircraft family (4000KS—electronic control unit/electronic engine unit, 4000HA—pressure bleed valve, and 438HC—trim air valve). For example, 4000HA with the highest imbalance ratio of 0.0047 has a false-negative rate of about 0.000639 compared to 4000KS with the lowest imbalanced ratio and false-negative rate of 0.00011. The analysis for A320 (11HB—flow control valve, 10HQ—Avionics equipment ventilation computer, 1TX1—air traffic service unit) shows insignificant changes to the imbalance ratio in terms of false-negative rate.

6 Conclusion and future work

This study presents a novel method for condensing a large number of logs of aircraft warning and failure messages recorded by the central maintenance system into a small number of the most significant and relevant logs. The reduced log is then used to create a model for aircraft' predictive maintenance, with an emphasis on predicting extremely infrequent failures. The proposed model combines an auto-encoder with bidirectional gated recurrent networks, which work together to deliver correct link failure/warning signals related to aircraft LRU removal while also assisting in the detection of abnormal patterns and trends. The auto-encoder is in charge of detecting unusual failures, while the BGRU networks (with CNN) are in charge of making predictions. The proposed technique is evaluated using real-world aircraft central maintenance system (ACMS) data. The evaluation results indicate that the AE–CNN–BGRU model can effectively handle irregular patterns and trends, mitigating the imbalanced classification problem. Comparing AE–CNN–BGRU with other similar deep learning methods, the proposed approach shows superior performance with 18% better precision, 5% in a recall, and 10% in g-mean. The results also indicate the model effectiveness in predicting component failure within a defined useful period that aids in minimising operational disruption. By traversing the input data in a bidirectional manner (feedforward and feedback) while making the prediction, the AE–CNN–BGRU model networks can better capture the underlying temporal structure. For specific types of data, such as in-text classification and text-to-words prediction in sequence-to-sequence learning, the performance advantage of AE–CNN–BGRU over the unidirectional GRU is reasonable. However, it was unclear whether employing a bidirectional strategy to train imbalanced time-series data would increase model performance because there may not be enough definite temporal contexts and observable in-text sequence examples. Our findings reveal that AE–CNN–BGRU outperforms standard GRU in forecasting

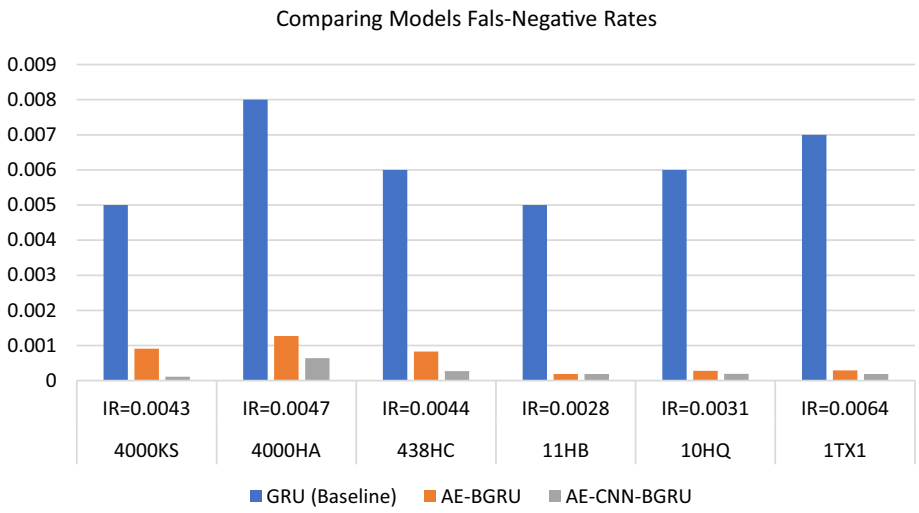


Fig. 17 Sensitivity analysis of imbalanced ratio against false-negative rate

uncommon failure in log-based aircraft ACMS datasets, answering this topic.

In the future, other AE–CNN–BGRU architectures will be studied further by translating time data into graphical representations utilising recurrence plots. The generated images can be trained with CNN-BGRU to improve their performance. Other aircraft data can also be imported into ACMS to improve model training.

Acknowledgements The authors would like to acknowledge the Integrated Vehicle Health Management Centre (IVHM), Cranfield University, and the first author would like to thank PTDF Nigeria for sponsoring his study.

Author contributions Maren designed, coordinated this research, and drafted the manuscript. Maren, Steve, and Zakwan carried out experiments and data analysis. Ian and Steve proofread and participated in research coordination. The authors read and approved the final manuscript.

Funding This study is funded by IVHM Centre, Cranfield University, UK, in relation to PTDF, Nigeria.

Availability of data and materials All softwares used for supporting the conclusions of this article are available in the public. The dataset used is confidential. It will be available based on request.

Declarations

Conflict of interest The authors declare that they have no competing interests. The authors have no relevant financial or non-financial interests to disclose.

References

1. Dangut MD, Skaf Z, Jennions IK (2020) Rare failure prediction using an integrated auto-encoder and bidirectional gated recurrent unit network. IFAC-PapersOnLine 53:276–282. https://doi.org/10.1016/j.ifacol.2020.11.045
2. Kingsley-Jones M. (2017) Airbus sees big data delivering "zero-AOG" goal within 10 years. Flightglobal
3. Wang Y. (2018) Strategies for aircraft using model-based prognostics
4. Buijs YJ. (2018) Integration of smart maintenance and spare part logistics for healthcare systems
5. Krawczyk B (2016) Learning from imbalanced data: open challenges and future directions. Prog Artif Intell 5:221–232. https://doi.org/10.1007/s13748-016-0094-0
6. Dangut MD, Skaf Z, Jennions I (2020) Aircraft predictive maintenance modeling using a hybrid imbalance learning approach. SSRN Electron J. https://doi.org/10.2139/ssrn.3718065
7. Raghuwanshi BS, Shukla S (2018) UnderBagging based reduced Kernelised weighted extreme learning machine for class imbalance learning. Eng Appl Artif Intell 74:252–270. https://doi.org/10.1016/j.engappai.2018.07.002
8. Wu Z, Lin W, Ji Y (2018) An integrated ensemble learning model for imbalanced fault diagnostics and prognostics. IEEE Access 6:8394–8402. https://doi.org/10.1109/ACCESS.2018.2807121
9. Zhang Y, Li X, Gao L, Wang L, Wen L, Lee DH et al (2020) Deep learning for smart manufacturing: methods and applications. J Manuf Syst 56:1–13. https://doi.org/10.1016/j.jmsy.2018.01.003
10. Blancke O, Combette A, Amyot N, Komljenovic D, Lévesque M, Hudon C et al (2018) A predictive maintenance approach for complex equipment based on petri net failure mechanism propagation model. Proc Eur Conf PHM Soc 4:1–12
11. Blancke O, Komljenovic D, Tahan A, Combette A, Amyot N, Lévesque M, et al. (2018) A predictive maintenance approach for complex equipment based on petri net failure mechanism propagation model. In: Proc Eur Conf PHM Soc p. 1
12. Aivaliotis P, Georgoulias K, Arkouli Z, Makris S (2019) Methodology for enabling digital twin using advanced physics-based modelling in predictive maintenance. Procedia CIRP 81:417–422. https://doi.org/10.1016/j.procir.2019.03.072
13. Parris CJ. (2016) The future for industrial services - the digital twin. Infosys Insights pp. 42–9
14. Okoh C, Roy R, Mehnen J (2017) Predictive maintenance modelling for through-life engineering services. Procedia CIRP 59:196–201. https://doi.org/10.1016/j.procir.2016.09.033
15. Phillips P, Diston D (2011) A knowledge driven approach to aerospace condition monitoring. Knowledge-Based Syst 24:915–927. https://doi.org/10.1016/j.knosys.2011.04.008
16. Ferri FAS, Rodrigues LR, Gomes JPP, De Medeiros IP, Galvao RKH, Nascimento CL. (2013) Combining PHM information and system architecture to support aircraft maintenance planning. In: SysCon 2013 - 7th Annu IEEE Int Syst Conf Proc pp. 60–5. Doi: https://doi.org/10.1109/SysCon.2013.6549859
17. Berberidis C, Angelis L, Vlahavas I (2004) Inter-transaction association rules mining for rare events prediction. In: Proc 3rd Hell Conf
18. Wu Z, Guo Y, Lin W, Yu S, Ji Y (2018) A weighted deep representation learning model for imbalanced fault diagnosis in cyber-physical systems. Sensors (Switzerland). https://doi.org/10.3390/s18041096
19. Nguyen KTP, Medjaher K (2019) A new dynamic predictive maintenance framework using deep learning for failure prognostics. Reliab Eng Syst Saf 188:251–262. https://doi.org/10.1016/j.ress.2019.03.018
20. Johnson JM, Khoshgoftaar TM (2019) Survey on deep learning with class imbalance. J Big Data. https://doi.org/10.1186/s40537-019-0192-5
21. Pouyanfar S, Tao Y, Mohan A, Tian H, Kaseb AS, Gauen K, et al. (2018) dynamic sampling in convolutional neural networks for imbalanced data classification. In: Proc. - IEEE 1st Conf. Multimed. Inf. Process. Retrieval, MIPR 2018, p. 112–7. Doi: https://doi.org/10.1109/MIPR.2018.00027
22. Lee H, Park M, Kim J. (2016) Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In: Proc - Int Conf Image Process ICIP 2016; -Augus, pp. 3713–7 doi: https://doi.org/10.1109/ICIP.2016.7533053

23. Wang S, Liu W, Wu J, Cao L, Meng Q, Kennedy PJ. (2016) Training deep neural networks on imbalanced data sets. In: Proc Int Jt Conf Neural Networks 2016-Octob, pp. 4368–74. https://doi.org/10.1109/IJCNN.2016.7727770
24. Buda M, Maki A, Mazurowski MA (2018) A systematic study of the class imbalance problem in convolutional neural networks. Neural Netw 106:249–259. https://doi.org/10.1016/j.neunet.2018.07.011
25. Song J, Shen Y, Jing Y, Song M. (2017) Towards deeper insights into deep learning from imbalanced data 2: 674–84. https://doi.org/10.1007/978-981-10-7299-4_56
26. Hensman P, Masko D. (2015) The impact of imbalanced training data for convolutional neural networks. PhD
27. Pouyanfar S, Tao Y, Mohan A, Tian H, Kaseb AS, Gauen K, et al. (2018) Dynamic sampling in convolutional neural networks for imbalanced data classification. In: Proc - IEEE 1st Conf Multimed Inf Process Retrieval, MIPR 2018, pp. 112–7 doi: https://doi.org/10.1109/MIPR.2018.00027.
28. Khan SH, Hayat M, Bennamoun M, Sohel FA, Togneri R (2018) Cost-sensitive learning of deep feature representations from imbalanced data. IEEE Trans Neural Netw Learn Syst 29:3573–3587. https://doi.org/10.1109/TNNLS.2017.2732482
29. Zhang C, Tan KC, Ren R. (2016) Training cost-sensitive Deep Belief Networks on imbalance data problems. In: Proc. Int. Jt. Conf. Neural Networks, vol. 2016- Octob, p. 4362–7. Doi: https://doi.org/10.1109/IJCNN.2016.7727769
30. Wang H, Cui Z, Chen Y, Avidan M, Ben AA, Kronzer A (2018) Predicting hospital readmission via cost-sensitive deep learning. IEEE/ACM Trans Comput Biol Bioinf. https://doi.org/10.1109/TCBB.2018.2827029
31. Lin TY, Goyal P, Girshick R, He K, Dollar P (2017) Focal loss for dense object detection. Proc IEEE Int Conf Comput Vis. https://doi.org/10.1109/ICCV.2017.324
32. Keisuke Nemoto , Ryuhei Hamaguchi , Tomoyuki Imaizumi SH. Classification of rare building change using cnn with multi-class focal loss Keisuke Nemoto , Ryuhei Hamaguchi , Tomoyuki Imaizumi , Shuhei Hikosaka Satellite Business Division , PASCO CORPORATION (Japan) 2018:4667–70
33. Salfner F, Lenk M, Malek M (2010) A survey of online failure prediction methods. ACM Comput Surv. https://doi.org/10.1145/1670679.1670680
34. Zhang K, Xu J, Min MR, Jiang G, Pelechrinis K, Zhang H. (2016) Automated IT system failure prediction: a deep learning approach. In: Proc - 2016 IEEE Int Conf Big Data, Big Data 2016, pp. 1291–300. https://doi.org/10.1109/BigData.2016.7840733
35. Korvesis P, Besseau S, Vazirgiannis M. (2018) Predictive maintenance in aviation: Failure prediction from post-flight reports. In: Proc - IEEE 34th Int Conf Data Eng ICDE 2018, pp. 1423–34. Doi: https://doi.org/10.1109/ICDE.2018.00160
36. Sipos R, Wang Z, Moerchen F. (2014) Log-based predictive maintenance, pp. 1867–76
37. Kallas M, Mourot G, Anani K, Ragot J, Maquin D (2017) Fault detection and estimation using kernel principal component analysis. IFAC-PapersOnLine 50:1025–1030. https://doi.org/10.1016/j.ifacol.2017.08.212
38. Lee J-M, Qin SJ, Lee I-B (2008) Fault detection of non-linear processes using kernel independent component analysis. Can J Chem Eng 85:526–536. https://doi.org/10.1002/cjce.5450850414
39. Ismail Fawaz H, Forestier G, Weber J, Idoumghar L, Muller PA (2019) Deep learning for time series classification: a review. Data Min Knowl Discov 33:917–963. https://doi.org/10.1007/s10618-019-00619-1
40. Guo S, Yang T, Gao W, Zhang C (2018) A novel fault diagnosis method for rotating machinery based on a convolutional neural network. Sensors (Switzerland). https://doi.org/10.3390/s18051429
41. Park P, Di Marco P, Shin H, Bang J (2019) Fault detection and diagnosis using combined autoencoder and long short-term memory network. Sensors (Switzerland) 19:1–17. https://doi.org/10.3390/s19214612
42. Liu R, Yang B, Zio E, Chen X (2018) Artificial intelligence for fault diagnosis of rotating machinery: a review. Mech Syst Signal Process 108:33–47. https://doi.org/10.1016/j.ymssp.2018.02.016
43. Dangut MD, Skaf Z, Jennions IK (2020) An integrated machine learning model for aircraft components rare failure prognostics with log-based dataset. ISA Trans 113:127–139. https://doi.org/10.1016/j.isatra.2020.05.001
44. Burnaev E. (2019) Rare failure prediction via event matching for aerospace applications. In: 2019 3rd Int Conf Circuits, Syst Simulation, ICCSS 2019, pp. 214–20. https://doi.org/10.1109/CIRSYSSIM.2019.8935598
45. Che C, Wang H, Fu Q, Ni X (2019) Combining multiple deep learning algorithms for prognostic and health management of aircraft. Aerosp Sci Technol 94:105423. https://doi.org/10.1016/j.ast.2019.105423
46. Burnaev E. (2019) Rare failure prediction via event matching for aerospace applications
47. Baldi P (2012) Autoencoders, unsupervised learning, and deep architectures. ICML Unsupervised Transf Learn. https://doi.org/10.1561/2200000006
48. Le Q V. A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks. Tutorial 2015:1–20
49. Farzad A, Gulliver TA. (2019) Log message anomaly detection and classification using auto-B/LSTM and auto-GRU, pp. 1–28
50. Konar A. (1999) Artificial intelligence and soft computing. https://doi.org/10.1201/9781420049138
51. Savoy J, Gaussier E. (2010) Information retrieval. https://doi.org/10.4324/9781351044677-24
52. Livieris IE, Pintelas E, Pintelas P (2020) A CNN–LSTM model for gold price time-series forecasting. Neural Comput Appl 32:17351–17360. https://doi.org/10.1007/s00521-020-04867-x
53. Debayle J, Hatami N, Gavet Y. (2018) Classification of time-series images using deep convolutional neural networks, 23 doi: https://doi.org/10.1117/12.2309486
54. Jafari G, Shirazi AH, Namaki A, Raei R. (2011) Coupled time series analysis: Methods and applications. vol. 13. Doi: https://doi.org/10.1109/MCSE.2011.102
55. Lu W, Li J, Wang J, Qin L (2020) A CNN-BiLSTM-AM method for stock price prediction. Neural Comput Appl 33:4741–4753. https://doi.org/10.1007/s00521-020-05532-z
56. Zhao B, Lu H, Chen S, Liu J, Wu D (2017) Convolutional neural networks for time series classification. J Syst Eng Electron 28:162–9. https://doi.org/10.21629/JSEE.2017.01.18
57. Ouhame S, Hadi Y, Ullah A (2021) An efficient forecasting approach for resource utilisation in cloud data center using CNN-LSTM model. Neural Comput Appl 33:10043–10055. https://doi.org/10.1007/s00521-021-05770-9
58. Munna MTA, Alam MM, Allayear SM, Sarker K, Ara SJF (2020) Prediction model for prevalence of type-2 diabetes complications with ANN approach combining with K-fold cross validation and K-means clustering, vol 69. Springer, Berlin
59. Applications C. Mathematical and computational applications, 2011;16:702–11.
60. Jiang P, Chen J (2016) Displacement prediction of landslide based on generalised regression neural networks with K-fold cross-validation. Neurocomputing 198:40–47. https://doi.org/10.1016/j.neucom.2015.08.118
61. David Dangut M, Skaf Z, Jennions I. (2020) Rescaled-LSTM for predicting aircraft component replacement under imbalanced

dataset constraint. In: 2020 Adv. Sci. Eng. Technol. Int. Conf. ASET 2020, doi: https://doi.org/10.1109/ASET48392.2020.9118253
62. Kamath U, Liu J, Whitaker J (2019). Deep Learning for NLP and Speech Recognition. https://doi.org/10.1007/978-3-030-14596-5
63. Lecun Y, Bottou L, Bengio Y, Ha P. (1998) LeNet. Proc IEEE, pp. 1–46
64. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521:436–444. https://doi.org/10.1038/nature14539
65. Bengio Y, Courville A, Vincent P (2013) Representation learning: a review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35:1798–1828. https://doi.org/10.1109/TPAMI.2013.50
66. David Dangut M, Skaf Z, Jennions I. (2020) Rescaled-LSTM for predicting aircraft component replacement under imbalanced dataset constraint. In: 2020 Adv. Sci. Eng. Technol. Int. Conf., IEEE; pp. 1–9. https://doi.org/10.1109/ASET48392.2020.9118253
67. Roc B. (2021) Comparing two ROC curves – independent groups design. NCSS, LLC, pp. 1–26