DETERMINATION OF REMAINING USEFUL LIFE

Sindhu Kaleeswaran
Final year student,
Department of Computational Intelligence,
SRM Institute of Science & Technology, Kattankulathur.
sk3454@srmist.edu.in

Chereddy Sowmya Sri
Final year student,
Department of Computational Intelligence,
SRM Institute of Science & Technology, Kattankulathur.
cq5256@srmist.edu.in

Dr.Jackulin Mahariba A
Assistant Professor,
Department of Computational Intelligence,
SRM Institute of Science & Technology, Kattankulathur.
jackulia@srmist.edu.in

Abstract—Life support system (LSS) is a complex part of the entire submarine that is designed to ensure crew members safety, sustained living, and survival during underwater missions. The OIS present in the LSS plays a major role in supplying breathable air to the crew through various Oxygen valves. The estimation of Remaining Useful Life (RUL) for OIS is accomplished using a survival estimator with a Bayesian Convolutional Neural Network (BCNN) model, providing a comprehensive approach to predict system survival. This is built by considering various ambient parameters such as Temperature, oxygen concentration, humidity and absolute pressure of the sphere and human parameters such as heart rate, body temperature, blood oxygen and blood pressure which is then displayed on the monitor that makes the crewmen aware of their present conditions. Thus, the OIS and LSS are integrated and provides time to time data on the RUL to the crew members. We anticipate that the proposed method can be widely applicable to estimate the RUL in OIS.

Keywords— Convolutional Neural Network, Remaining Useful life, Variational Inference, Bayesian convolutional neural network, Oxygen Injection System, Life support system, End of life, Fitness for Service

I.INTRODUCTION

RUL is the length of time a system is expected to work/sustain before it needs replacement. It refers to the remaining operational time of the system before it reaches a predefined failure threshold and operational duration left for the system before it becomes non-functional or requires maintenance. The RUL concept is crucial in predictive maintenance strategies for OIS, aiming to optimize system reliability, minimize downtime, and reduce operational costs. Finding the remaining useful life for a LSS is very important so that it increases the reliability of the mission. The RUL of LSS is calculated by continuously monitoring various operational parameters, such as oxygen flow rates, pressure levels, temperature, and other crucial system performances, to predict the RUL of the OIS. This approach helps in timely interventions in a mission, preventive maintenance actions and efficient resource allocation by ensuring the OIS operates at its optimal performance level. Incorporating RUL predictions into the maintenance and operational protocols of an OIS facilitates improved system reliability and enhanced safety for the system as well as the crew members. Estimating RUL in OIS is crucial for proactive maintenance scheduling, ensuring optimal system performance, and minimizing downtime by predicting when maintenance or replacement actions should be taken.

II.LITERATURE SURVEY

A new prognostic tool for mechatronic systems condition monitoring, emphasizing the enhancement of predictive accuracy for criteria like RUL and State of Health [1]. This approach employed Hidden Markov Models to identify decline dynamics from completely instrumented dimension sets gathered in exploration settings and applied virtual sensing to estimate real-time amounts from limited functional measures. The optimal idle state dimension aligned with the number of features in most cases. The paper envisioned future work expanding into broader prognostic operations. Details were provided regarding the design and manufacturing of a subsea solenoid stopcock prototype [2] for subsea blowout preventers. The research employed deterministic and probabilistic thermal and electromagnetic analyses through ANSYS software. Results highlighted the significant influence of specific design parameters on maximum temperature and electromagnetic force, emphasizing the importance of probabilistic analysis in optimizing subsea solenoid stopcock design for enhanced reliability in blowout preventers. Another study introduced a methodology utilizing BCNNs for direct prediction of RUL in SV [3] based on current signatures. The proposed BCNN approach significantly enhanced predictive accuracy, reducing RUL Mean Absolute Error by approximately 40% compared to previous methods involving Deep Learning and feature-based techniques. Further improvement was achieved by incorporating salient physical features from an electromechanical model, resulting in a hybrid model with a ∼20% lower RUL MAE. The BCNN architecture provided well-calibrated uncertainty estimations, crucial for reliable prognostic decision-making. Insights into the improved performance were gained through occlusion, a feature attribution method, underscoring the superiority of hybrid models for early RUL prediction. The Bayesian nature of the BCNN, utilizing Mean-Field Variational Inference architecture, ensured well-calibrated predictive uncertainty, facilitating trustworthy prognostic decisions. In another study, a data-driven approach for predicting RUL was explained. It employed an end-to-end combined autoencoder-regressor architecture [4] that combined long-short term memory networks and convolutional neural networks. The model, optimized using genetic algorithms, showed competitive performance in RUL prediction. Furthermore, a new defect detection-based RUL labeling technique was presented, exhibiting performance on par with or better than linear labeling. Non-linear adjustments were shown to boost model performance by up to 50% even with just one flaw in the sensor data. Improving architectural stability and fault detection-based RUL labeling will be the primary objective of future research. Finally, a novel approach for estimating the RUL of rolling bearings was proposed to overcome the problems of external uncertainties and real-time monitoring. It employed an enhanced dropout technique utilizing nonparametric kernel density and a fusion measure for runtime, based on LSTM [5] with uncertainty characterization. Probabilistic distribution and precise RUL point estimation were demonstrated via validation of the dataset. The method's practical significance was indicated by its robust performance, particularly during stable deterioration. Moreover, by assessing and choosing efficient degrading features, a novel evaluation index exhibited higher performance over random-feature-trained LSTM models in terms of accuracy and stability.

The Prognosis and Health Management in real-time monitoring and maintenance planning, providing a methodical approach [6] for ML model selection and development based on run-to-failure data. Their analysis encompasses contemporary ML technologies for RUL prediction, while also identifying areas for improving data quality and model reconstruction, along with highlighting avenues for further development. Another study introduces a simulation tool enabling online simulation of PHM systems, facilitating experimentation with different ML scenarios [7] for estimating RUL of aviation systems. The tool's user-friendly interface allows configuration of dataset size, RUL prediction techniques, and outcome metrics, while offering cutting-edge ML approaches. Future enhancements could expand its applicability through additional techniques, datasets, and model customization options. A novel approach for accurately predicting the residual life of rolling bearings, integrating elastic net regularization with LSTM [8] to enhance stability and forecasting performance. Ongoing research aims to optimize RUL prediction efficiency and computational speed using this method. Another study presents a Bayesian least-squares support vector machine (LS-SVM) [9] method for RUL prediction of a specific component, demonstrating superior accuracy and reliability compared to existing techniques. They emphasize the importance of refining model parameters and exploring suitable kernel functions to enhance accuracy and adaptability across diverse scenarios. Artificial neural networks are highlighted as the most accurate method for predicting gearbox failure in wind turbines, particularly when utilizing SCADA [10] and vibration data. Further algorithmic improvements and incorporation of additional SCADA inputs are suggested for more precise forecasts, indicating the potential of multiclass neural networks in conjunction with vibration data.

Several studies detail innovative techniques for predicting the RUL of critical components. One study introduces a method using Hilbert-Huang entropy [11] from horizontal vibration signals to track bearing degradation, employing a linear degradation model for RUL prediction. Another study employs artificial neural networks [12] to model pneumatic actuators, enhancing fault detection strategies. The analysis emphasizes the importance of actuator fault detection within broader fault detection and isolation systems. Additionally, research explores the connection between static and dynamic characteristics of pneumatic valves [13], aiming for more accurate dynamic measurements to assess valve performance. Another study proposes a Bayesian large kernel attention network [14] for RUL prediction, prioritizing feature selection and uncertainty quantification. The experiments validate its efficacy in quantifying uncertainty and outperforming baseline models. Furthermore, literature surveys indicate a growing interest in utilizing BCNNs [15] for predicting RUL in solenoid valves, emphasizing their potential for improving maintenance and reliability through enhanced predictive performance and calibrated uncertainty estimation.

A neural network model developed to predict piping lifespan based on real-time inspection data demonstrates high accuracy, indicating its potential to enhance FFS[16] assessments and reduce inspection costs and workforce demands for operators. The paper critiques existing deep learning-based RUL prediction models for neglecting operating condition data and overemphasizing stacking network layers. To address these issues, a novel multi-dimensional recurrent neural network (MDRNN) is proposed, integrating parallel bidirectional LSTM and GRU[17] pathways to capture diverse degradation features, effectively modeling multisensory monitoring and operating condition data simultaneously. Experimental validation on aircraft turbofan datasets showcases MDRNN's superior performance under variable operating conditions and multiple fault modes. The literature underscores the critical need for accurate gear remaining life prediction due to the severe consequences of gear failures in industrial equipment. A novel approach combines a health indicator fused from time-domain and frequency-domain vibration signal features via the ISOMAP algorithm with an advanced LSTM variant LSTMP-A. LSTMP-A[18] enhances prediction accuracy by amplifying input and recurrent weights based on an attention mechanism. Comparative experiments validate LSTMP-A's outperformance of conventional methods in prediction accuracy, with future research focusing on refining its structure and extending its application to variable working conditions. Additionally, the application of MDRNN for RUL prediction under variable operating conditions and multiple fault modes (VOCMFM) has shown promising results. By integrating parallel bidirectional LSTM and GRU pathways, MDRNN effectively models multisensory monitoring and operating condition data simultaneously. Validation on aircraft turbofan datasets confirms its superior performance compared to state-of-the-art models. Bayesian DL models further enhance RUL prediction by quantifying uncertainty, which is critical for applications where unplanned failures can lead to high costs or human harm. Experiments with Bayesian models using Hamiltonian Monte Carlo[19] and variational inference demonstrate improved predictive performance and reliability. Moreover, an EMD-CNN model for bearing RUL estimation leverages Empirical Mode Decomposition to handle nonstationary degradation signals, feeding the resultant Intrinsic Mode Functions into a CNN. This model, validated on the PHM2012 dataset[20], shows high prediction accuracy, especially when using a loss-based weighting ensemble method. Furthermore, a deep Bayesian survival approach named BNN-Surv[21] is proposed to handle censored data for rail useful lifetime modelling, incorporating Monte Carlo dropout to quantify uncertainty. Applied to a comprehensive railway dataset, BNN-Surv outperforms traditional models, offering both accurate predictions and valuable confidence intervals for better decision-making and maintenance planning. Lastly, a Bayesian deep-active-learning[22] framework addresses the challenge of limited run-to-failure data by integrating uncertainty quantification and active learning. This method significantly reduces the need for extensive run-to-failure data while maintaining high prediction accuracy, as demonstrated on bearing and battery datasets. Future work will explore the extension of this approach through transfer learning and its application to various industrial systems to enhance predictive maintenance strategies.

In addition, an inverse Gaussian process with random effects is proposed for adaptive RUL estimation, incorporating Bayesian updating to handle real-time degradation data. This method [23] effectively captures system heterogeneities and dynamically updates RUL predictions, demonstrating improved accuracy and reduced uncertainty in both numerical and practical case studies. A new symmetric directed divergence measure-based intelligent framework is introduced for bearing health assessment. It combines various techniques, including the Pauta criterion, adaptive wavelet filtering, novel health indicator development, and GCN-based RUL prediction [24], offering an integrated solution for degradation monitoring, defect identification, and RUL estimation. The framework's components, such as the alert system, adaptive wavelet filter, and deep learning regression, contribute to accurate RUL estimation even with limited training data. However, the framework's effectiveness is contingent upon high-quality training data, and generalization to diverse datasets remains a challenge, suggesting avenues for future research to enhance robustness and applicability. The Bayesian approach to parameter estimation in the failure forecast method contrasts it with the traditional linear regression formulation [25]. By relaxing the linearity assumption of the inverse feature rate, the Bayesian model enables the generation of probability distributions for failure forecasts, particularly beneficial for positive-feedback mechanisms like fatigue crack propagation. Comparative analyses on synthetic data and real fatigue experiments demonstrate the Bayesian model's superior convergence to true values of failure forecasts across varying noise levels, outperforming linear regression models. Additionally, the incorporation of an adaptive switch point in the Bayesian model enhances accuracy, suggesting future research directions in data windowing and addressing correlations between model parameters.

III.PROPOSED METHOD

This paper employs residual and hashed similarity-based transfer learning approach to predict the RUL of LSS. The transfer learning acquires the pre-trained BCNN model's knowledge and applying it to a new/target BCNN model, for a similar task, with minimal modifications to the existing model's architecture. This allows the learned features from the pre-trained model to effectively transfer them to the new RUL prediction task. Regression estimates continuous variables, for calculating RUL of the system. The Bayesian CNN, which is employed in regression tasks, captures complex relationships and patterns in the data to make accurate and reliable predictions. While the pre-trained model utilized a BCNN, the target model incorporated a Bayesian VI layer specifically for uncertainty estimation. The core architecture of the BCNN in the pre-trained model remained largely unchanged, with the only modification being the replacement of its final dense layer with a Bayesian VI layer in the target model. This addition enables that the knowledge and information learnt from the pre-trained task are applied to the target task. By doing so, it enhances the model's predictive accuracy and reliability.

IV.DATA

Analysing the behaviour of the system, particularly in relation to current signal patterns, yields valuable insights regarding the condition of the system and potential EOL situations. These insights can be used to develop predictive models that estimate the RUL of a system. By detecting changes in the point to the transition from a healthy to a defective regime and ultimately to EOL, proactive maintenance strategies can be planned and executed in a timely manner. In the estimation of RUL, a comprehensive approach is undertaken, by taking few sets of parameters into consideration. These include various ambient parameters [17] such as temperature, oxygen concentration, humidity and absolute pressure of the sphere, measured using installed sensors, as well as human parameters such as heart rate, body temperature, blood oxygen and blood pressure obtained through wearable sensors. These parameters were transformed into visual representations, where grayscale intensity corresponds to the magnitude of each parameter value. By allocating a scale on the grayscale spectrum for each parameter, these visualizations provide a comprehensive overview of the variables. To effectively categorize and interpret the data, clustering is employed to segregate the parameters into three distinct classes Fig 1 based on their characteristics.

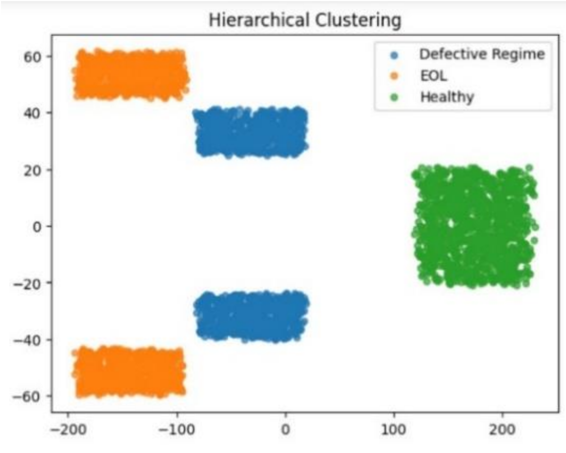

Fig 1 Hierarchical Clustering of End of life,Defective Regime and Healthy states of the sphere

Through hierarchical clustering, a tree-based structure known as a dendrogram Fig 2, displays the cluster arrangement of three classes stating the healthy, defective regime and EOL of the sphere. Each branch of the dendrogram represents a cluster, with the length of the branches indicating the dissimilarity between clusters.

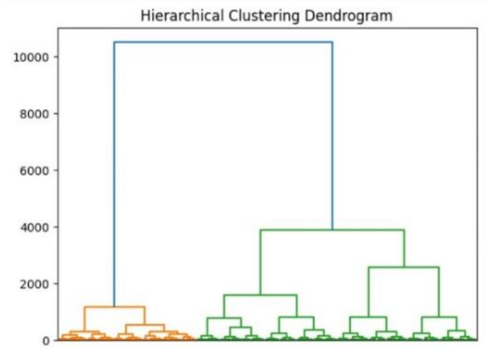

Fig 2 Visual representation of merged clusters

Through the utilization of these images, a clear comprehension emerges regarding the system's prediction on longevity efficiency.

V.METHODOLOGY

A BCNN[3][6] is a type of neural network that incorporates an inference based on statistical analysis for the purpose of determining uncertainty. Point estimates of their predictions or parameters are often provided by traditional neural networks such as a feedforward or convolutional neural network.

A. Bayesian convolutional neural network

BCNN is an extension of traditional CNN that incorporates components such as

I.Bayesian convolutional layers: In a BCNN, the convolutional layers are inherently Bayesian. This means that, rather than having fixed weights and biases, each filter in the convolutional layers is represented by a probability distribution of its parameters. These distributions such as mean and variance, are learned during training using Bayesian inference techniques.

II.Parameter Uncertainty: The Bayesian treatment of parameters introduces uncertainty into the network. A probability distribution is used to represent each parameter instead of a single set. This expresses the model's uncertainty about the true values of parameters based on observed data.

III.Variational Inference: This is a common technique for approximating the posterior distributions of Bayesian layers parameters. During training, variational inference entails minimizing the divergence measure between the approximate and the true posterior distributions.

IV.Monte Carlo Dropout**:** Used in BCNNs to approximate Bayesian inference. During inference, dropout is applied to the network, and multiple forward passes are performed with dropout enabled. Predictions are then made by averaging or integrating the predictions from multiple forward passes, resulting in an estimate of uncertainty.

V.Loss Function: During BCNN training, this is typically used to minimize prediction error.

B. Transfer Learning

Transfer learning is employed to enhance the performance of the target model in predicting the RUL. This approach enables the model to adapt its learned representations to the new task more efficiently. Specifically, a pre-trained model is utilized, where all layers except for the last four are set as non-trainable to retain learned features. The final layer is then replaced with a Bayesian DenseVI layer to facilitate Bayesian inference and uncertainty estimation. Successively, a custom loss function incorporating mean absolute error and KL divergence regularization is defined to ensure robust uncertainty estimates. Through optimization with the Adam optimizer and early stopping based on validation performance, the target model undergoes fine-tuning on the scaled training data for predicting RUL. After preprocessing, the model weights are obtained and converted into an image. This image provides an insight into what the network has learned and visualizing the weights can help understand which features or patterns the model is focusing on during training.

The weights of the convolutional and dense layers are adjusted iteratively through backpropagation to minimize the loss function and optimizing the model for the task of predicting RUL. During transfer learning, the weights of the last few layers of the pre-trained model are modified and fine-tuned to adapt its learned features to the specifics of the RUL prediction problem.

C. KL Divergence

The KL divergence is a measure of how one probability distribution diverges from a second probability distribution. It quantifies the difference between two probability distributions, P and Q. The KL divergence is not symmetric and is always non-negative. It reaches zero if and only if P and Q are identical. In the context of reliability, the KL divergence can be used, for calculating, RUL using BCNN [20] for set of parameters 41 considered in this research work.

$$D_{KL}(P||Q) = \int (x) \log(\frac{P(x)}{Q(x)}) dx \text{..............................Eqn 1}$$

After specifying the prior and likelihood, Bayes theorem is used to generate the posterior distribution over the model weights,

$$\pi(\omega|D) = \frac{p(\omega)p(D|\omega)}{\int p(\omega)p(D|\omega)d\omega} = \frac{p(\omega)p(D|\omega)}{p(D)} \text{..................Eqn 2}$$

Predictions are in the form of an expectation with respect to the posterior distribution,

$$\mathbb{E}_{\pi}[f] = \int f(\omega)\pi(\omega|D)d\omega \text{..............................Eqn 3}$$

where, ω is the set of parameters in the generic statistical model, D is our data and H_i represents the i-th model used for this level of inference. This is then described as

$$(\omega|D,\mathcal{H}_i) = \frac{P(D|\omega,\mathcal{H}_i)P(\omega|\mathcal{H}_i)}{P(D|\mathcal{H}_i)} \text{.........................Eqn 4}$$

Posterior = (Likelihood*Prior)/Evidence……..………Eqn 5

Assuming a Gaussian distribution, the Laplace approximation of the evidence is found as,

$$P(D|\mathcal{H}_i) = \int P(D|\omega,\mathcal{H}_i)P(\omega|\mathcal{H}_i)d\omega$$
$$\approx P(D|\omega_{MAP},\mathcal{H}_i)[P(\omega_{MAP}|\mathcal{H}_i)\Delta\omega]$$
$$= P(D|\omega_{MAP},\mathcal{H}_i)\left[P(\omega_{MAP}|\mathcal{H}_i)(2\pi)^{\frac{k}{2}}det^{\frac{-1}{2}}\mathrm{A}\right]$$

= Best Likelihood Fit x Occam Factor…………………Eqn 6

The Occam factor can be interpreted as the ratio of the width of the posterior $\Delta\omega$ and the range of the prior $\Delta\omega_0$ for the given model H_i

$$Occam\ Factor = \frac{\Delta\omega_0}{\Delta\omega} \text{.....................Eqn 7}$$

meaning that the Occam factor is the ratio of change in plausible parameter space from the prior to the posterior.

The assumed posterior distribution q (ω) is a suitable density over the set of parameters ω, that is restricted to a certain family of distributions parameterized by theta. The parameters of this variational distribution are adjusted to reduce the discrepancy between the variational distribution and the true posterior p(ω | D). The forward KL-Divergence between the variational and true distribution is frequently used to assess VI similarity,

$$KL(q_\theta(\omega)||p(\omega|D)) = \int q_\theta(\omega)\log\frac{q_\theta(\omega)}{p(\omega|D)}d\omega \text{.........Eqn 8}$$

For VI, the above equation serves as the objective function we wish to minimize w.r.t variational parameters. This can be expanded out as,

$$KL(q_\theta(\omega)||p(\omega|D)) = \mathbb{E}_q \log\frac{q_\theta(\omega)}{p(\omega)} - \log p(D|\omega) + \log p(D)$$
$$= KL(q_\theta(\omega)||p(\omega)) - \mathbb{E}_q[\log p(D|\omega)] + \log p(D).$$
$$= -\mathcal{F}[q_\theta] + \log p(D) \text{...Eqn 9}$$

Where, $\mathcal{F}[q_\theta] = -KL(q_\theta(\omega)||p(\omega)) + \mathbb{E}_q[\log p(D|\omega)]$

Here, the first term $KL[q_\theta(\omega)||p(\omega)]$ represents the negative KL divergence between the variational distribution and the true prior distribution p(θ), and the second term $E_q[\log p(D|\omega)]$ represents the expected log likelihood of the data under the variational distribution. The term $F[q_\theta]$ represents the ELBO and is defined as the sum of the expected log likelihood of the data under the variational distribution and the negative KL divergence between the variational distribution and the true posterior distribution. The goal is to maximize the ELBO with respect to the parameters of the variational distribution. Maximizing the ELBO encourages the variational distribution to be close to the true posterior distribution while maximizing the likelihood of the observed data.

ELBO can be broken down into two terms: data likelihood and KL divergence between the approximate posterior and the prior. The KL loss regularization serves as a penalty term inthe loss function to ensure that the learnt posterior distribution does not diverge significantly from the prior distribution. By doing so, the model is encouraged to find a posterior that strikes a balance between fitting the data accurately and maintaining simplicity by aligning it with the prior distribution. The KL loss regularization and the KL divergence with ELBO are related as they both serve to constrain the learned posterior distribution of the model parameters to be close to a given prior distribution. This constraint is crucial for ensuring the robustness and generalization capability of Bayesian Neural Networks by preventing overfitting and promoting model simplicity.

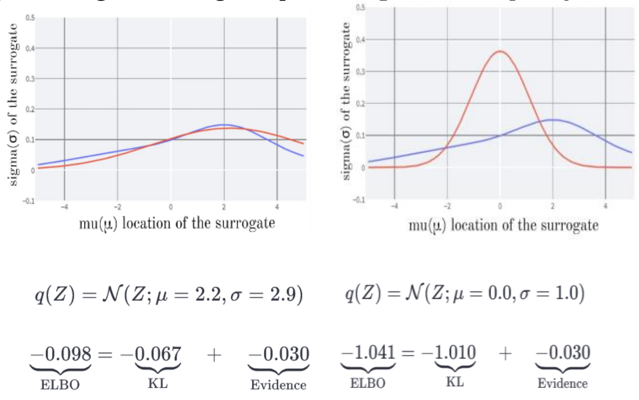

Fig 4a & 4b ELBO plots of variational and true posterior distribution (Adopted from [22])

The above pictures compare the ELBO plots of two distributions Fig 4a and 4b. X-axis denotes the mu(μ) location of the surrogate and the Y-axis denotes sigma(σ) of the surrogate. The blue line denotes the true posterior and red line denoted Surrogate. As discussed, the maximum the ELBO, the closer the distributions are.

D. Architecture of the models

This section outlines the architecture and methodology of the model's for predicting RUL.By associating each image with a distinct RUL value Fig 5, the predictive model gains insights into the system's longevity efficiency. Load these images with their respective RUL from preprocessing module, resizing them to a specified target size of (100, 100) pixels and normalizing the pixel values to the range [0, 1] by dividing each pixel value by 255. This normalization aids in improving the convergence and stability of the training process.

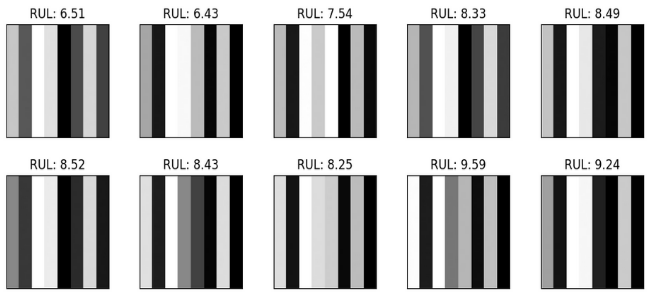

Fig 5 A random sample of ten images from the dataset along with their respective rul.

I. Bayesian Convolutional Layer

BCNNs, like standard CNNs, are composed of convolutional layers that apply filters to input data. These filters are trained to extract features from input images. Each filter identifies different patterns or features in the data.

II. Flatten Layer

It converts multidimensional data to a one-dimensional vector, typically required when switching from convolutional layers, which operate on multidimensional data such as images, to fully connected layers, which only accept one-dimensional input.

III. Pooling & Dropout Layer

In Bayesian pooling, uncertainty estimates are propagated through pooling layers by considering distributions over pooling operations. Setting dropout rate at 0.5, it serves as a regularization technique to prevent overfitting by randomly setting 50% of the input units to 0 during each training iteration. This dropout mechanism facilitates the network to learn more robust and generalized features.

IV. Dense Layer

Dense layers compute the weighted sum of inputs, add a bias term, and apply an activation function. Weights and biases in dense layers are learned during training, indicating the strength of connections between neurons and influencing the neuron's output.

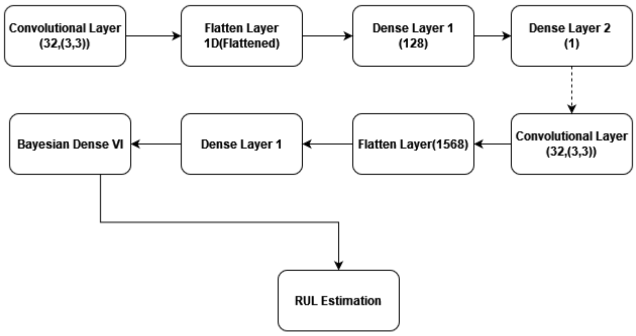

Fig 6 Architecture of Pre-trained BCNN model

We employ transfer learning using a pre-trained BCNN model for predicting RUL Fig 6. The pre-trained model architecture is modified in the target model by replacing the last layer with a new BayesianDenseVI layer with a single unit for regression. Each weight parameter in this layer is represented by two learnable parameters: the mean and the log of the standard deviation (rho).

During the forward pass, random noise sampled from a normal distribution is added to the mean parameter, scaled by the exponential of the rho parameter, introducing stochasticity into the layer, allowing for uncertainty estimation in the model's predictions. The model is compiled with a custom loss function incorporating MAE and KL divergence regularization.

Optimizer and early stopping callbacks are configured, and the model is trained on pre-processed training data while validating on a separate validation set to prevent overfitting. Early stopping is employed to prevent overfitting, and the training history is stored for analysis.

VI.IMPLEMENTATION

In this section Fig 7, two powerful methodologies are used: BayesianCNNs and transfer learning. The following steps are incorporated in RUL prediction.

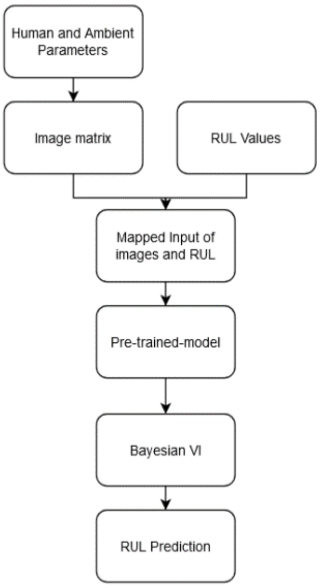

Fig 7 Flowchart for computing RUL

1.Parameters to image matrix

Initially, the data is read into a dataframe containing human and ambient parameter values. For each row in the dataframe, an intensity plot is generated using matplotlib, representing intensity with the x-axis denoting the row number and the y-axis denoting a fixed range of intensity for respective values.

2.Mapping images and RUL

Images are mapped with their respective RUL values by synchronizing time-based image filenames with RUL data.

3.Pre-Processing the data

For each image, it is loaded and resized to a specified target size of 100 x 100 pixels. The images are converted into numerical array representations and normalized by dividing the pixel values by 255. To expedite computations, the images are transformed to grayscale.

4.Bayesian CNN Architecture/ Pre-trained model

The BCNN model comprises convolutional, dense, dropout, and output layers. The model architecture includes a convolutional layer with 32 kernels of size 3 x 3 with ReLU activation, followed by a flattening layer and a dense layer of 128 units with ReLU activation. Dropout regularization technique is used with dropout rate about 0.5 to prevent over fitting. The final layer is the output layer with a single output, that gives out the final regression prediction.

5.Compilation and Training of the pre-trained model

The model is compiled using Adam optimizer with a learning rate of 0.001 and MAE as the loss function. Mini-batches of size 16 are processed during training for 10 epochs, with 20% of the training data allocated for validation.

6.Bayesian Dense VI Layer Definition/ Target model

This model Fig 8 implements the Bayesian inference techniques to model uncertainty in layer weights. This is done by the layer that introduces uncertainty into the model's weights by representing them as probabilistic distributions instead of fixed values. Within this layer, two sets of parameters, 'mean' and 'rho', are initialized. The 'mean' represents the mean values of the weights, while 'rho' captures the uncertainty or variance associated with these weights. During the forward pass, the model adds a bit of randomness (ε or epsilon) to its weights. This randomness comes from a normal distribution, which is like a bell-shaped curve of possible values. This allows the model to estimate the uncertainty in predictions and confidence level.

```
class BayesianDenseVI(Layer):
    def __init__(self, units, activation='relu'):
        super(BayesianDenseVI, self).__init__()
        self.units = units
        self.activation = tf.keras.activations.get(activation)
        self.batch_norm = BatchNormalization()

    def build(self, input_shape):
        self.mean = self.add_weight(shape=(input_shape[-1], self.units),
                                    initializer='random_normal',
                                    trainable=True,
                                    name='mean')
        self.rho = self.add_weight(shape=(input_shape[-1], self.units),
                                   initializer='zeros',
                                   trainable=True,
                                   name='rho')

    def call(self, inputs):
        epsilon = tf.random.normal(shape=tf.shape(self.mean))
        weights = self.mean + tf.math.log(1 + tf.exp(self.rho)) * epsilon
        x = tf.matmul(inputs, weights)
        x = self.batch_norm(x)
```

Fig 8 Architecture for BCNN model

7.Loading Pre-trained Model

Load a pre-trained BCNN model previously saved by using the function 'load_model' from TensorFlow Keras. This pre-trained model embodies a Bayesian CNN model trained earlier, acquiring features and representations crucial for the original task. By loading this model, computational efficiency is improved by utilizing the learned knowledge.

8.Freezing Pre-trained Model Layers

Freezing certain layers of the pre-trained model prevents them from being altered during training on new data. This helps to retain the knowledge learnt by the pre-trained model and preventing it from forgetting previously learned features while adapting to new ones.

9.Integration of Bayesian Dense VI Layer

The last layer of the pre-trained model is replaced with a Bayesian Dense VI layer, marking the integration of Bayesian uncertainty estimation into the pre-trained model. This layer computes both prior and posterior distributions over layer weights using variational inference.

10.Define KL Divergence Regularizer

A function is defined to calculate the KL divergence regularization term for the Bayesian Dense VI layer. This term enables the learned weights to approximate a prior distribution using Gaussian.

11.Custom Loss Function Definition with KL Divergence

A custom loss function is defined that combines the original loss which is the MAE between the true and predicted values along with the KL divergence regularization term. The regularization term penalizes deviations of learned weights from the prior distribution.

12.Compile Target Model

Compile the target model using the Adam optimizer with an even lower learning rate of 0.0001 for fine-tuning. The custom loss function, along with monitoring MAE loss metrics, is employed in the compilation process to optimize the model for RUL prediction.

13.Training the Target Model

The target model is fitted on the new training data and train for up to 50 epochs with early stopping to prevent overfitting. This monitors the performance on a validation dataset and stops the training process if the performance fails to improve for a specified number of consecutive epochs.

14.Obtaining RUL

Once the training of target model is done, the saved model is utilised for prediction of RUL. The predicted RUL value is then rounded to one decimal place and converted from a decimal representation to a time format in hours and minutes.

I. Compilation and Training

Compilation refers to the configuration of the learning process. When a model is compiled, the loss function, the optimizer, and the metrics that the model should use during training are specified. Training is the process of fitting the model to the training data. During training, the model adjusts its weights based on the optimization algorithm and the loss function to minimize the difference between the predicted RUL and the actual RUL. The performance metrics to assess the BCNN model is given below:

A. ***Mean Square Error (MSE)****:* It is a common metric used in regression problems to assess the performance of a predictive model. It is expressed as the average of the squared differences between predicted and actual values. The formula for MAE is defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \text{.....................Eqn 8}$$

Where *n* is the number of samples in the dataset. y_i is the true value of the target variable for the *i*-th sample. $\hat{y}$ is the predicted value of the target variable for the i-th sample. The evaluation of two metrics namely MSE with Stochastic CNN and MSE with BCNN are compared below:

• ***MSE with stochastic CNN:***

In a Stochastic CNN with Monte Carlo Dropout (MCDropout), MSE was initially used as a loss function to train the model. The stochastic nature of MCDropout introduces randomness, enabling the model to estimate uncertainty and optimize towards minimizing the MSE. However, the high variability in the loss made it less suitable. A visual representation Table 1 of an epoch in the regression neural network training process, demonstrating the progression of MSE loss over multiple iterations. To address this, another approach is employed to capture better uncertainty and optimize the model's performance.

Epochs	Loss	Validation Loss
1	58317781817104451054338 0.0000	451781.7263
2	404751270438854721536.0000	51821.1416
3	280915479910940672.0000	19132.9910
4	1949677225146112.0000	9476.8327
5	135316201472.0000	4397.7672
6	93915216.0000	2372.1936
7	65180.2461	937.9126
8	51.8937	19.1414
9	6.8616	2.8309
10	6.8313	2.8217

Table 1 Training Epoch for stochastic CNN + MSE

• ***MSE with BCNN:***

Continuing from the use of a Stochastic CNN with MSE, Bayesian CNN with MSE Table 2 was employed to better handle the uncertainty estimation. While the loss improved, it did not decrease significantly. So that, we need for further optimization for exploring different loss functions to better leverage the Bayesian framework and improve the model's performance.

Epochs	MSE Loss	MSE validation loss
1	25.1288	3.2606
2	6.3435	2.5490
3	6.0951	2.6409
4	6.0231	1.8956
5	6.0021	2.5387
6	5.7366	2.1936
7	5.5222	2.4657
8	5.8763	2.7271
9	5.5548	1.9433
10	5.4814	2.4343

Table 2 Training Epochs for BCNN +MSE

B. ***Mean Absolute Error (MAE):*** MAE is calculated by averaging the absolute differences between predicted and actual values. The formula is mentioned below:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\text{Eqn 9}$$

Where n is the number of samples in the dataset. y_i is the true value of the target variable for the i-th sample. $\hat{y}$ is the predicted value of the target variable for the i-th sample. The evaluation of MAE metrics with BCNN is given below:

• ***BCNN with MAE***

After experimenting with various approaches, the BCNN with MAE as the loss function proved to be the most effective Table 3, leading to minimal loss and superior performance in the task.

Epochs	MAE Loss	MAE validation loss
1	1.9695	1.3384
2	1.8780	1.1551
3	1.5743	1.2713
4	1.7081	1.2243
5	1.6353	1.1197
6	1.5846	1.0934
7	1.4962	1.1038
8	1.5325	1.1020
9	1.3987	1.1137
10	1.2378	1.0788

Table 3 Training Epoch for BCNN +MAE

VII.RESULT

A. Loss Prediction

loss graphs for both models provide useful information about their training methods and performance Fig 9. The MAE loss for the Bayesian CNN model decreases steadily during the first epochs, indicating that the training data is being learned effectively. However, as the epochs progress, the validation MAE may fluctuate or plateau, indicating potential overfitting or problems with generalization. On the other hand, the target model with Bayesian DenseVI[21] layer shows a different pattern. The MAE loss exhibits an initial decrease, similar to the Bayesian CNN. Still, with the addition of the Bayesian DenseVI layer and fine-tuning of the pre-trained model, the model is able to better leverage the learned features, potentially leading to more stable validation MAE values than the Bayesian CNN. Furthermore, the inclusion of the KL divergence loss component in the Transfer Learning model results in a regularization effect. This KL loss component aims to bring the Bayesian weights closer to a prior distribution, potentially enhancing the model's uncertainty quantification and robustness. In conclusion, while both models show a decrease in MAE loss over epochs, the Transfer Learning model's integration of the BayesianDenseVI layer and KL divergence regularization suggests a potentially more stable learning process with improved uncertainty quantification.

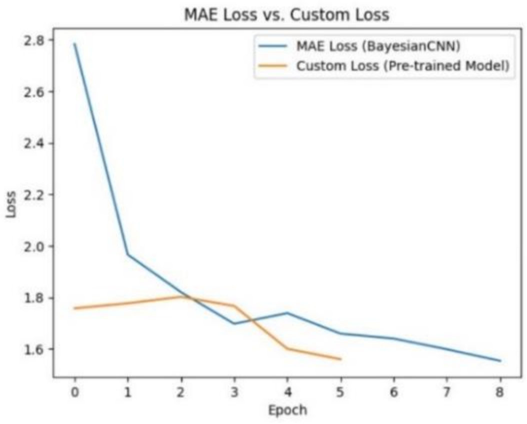

Fig 9 MAE vs Custom Loss

B. RUL Prediction

Once the training of target model is done, the saved model can be used for prediction of RUL which is displayed in the Fig 10a &10b &10c.The model's predictions are processed, rounding the RUL value to one decimal place for precision. Subsequently, this value is transformed from a decimal representation into a time format in hours and minutes. This final prediction helps in critical insights into the lifespan of submarine. Such accurate forecasting is necessary for ensuring the reliability and enhancing safety of submarine and its crew members.

Each parameter in x-axis is assigned a specific index which represents human and ambient parameters and the values of y-axis represents intensity of each parameter.

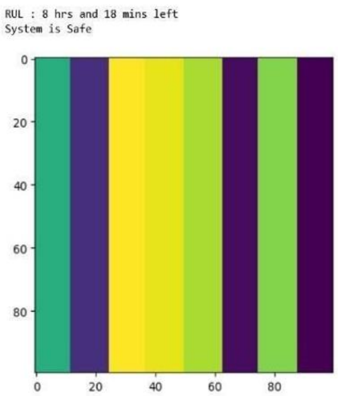

Fig 10a Predicted RUL for Healthy condition

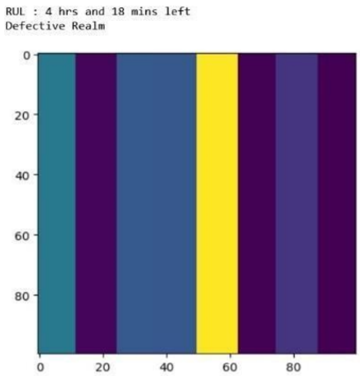

Fig 10b Predicted RUL for Defective Realm

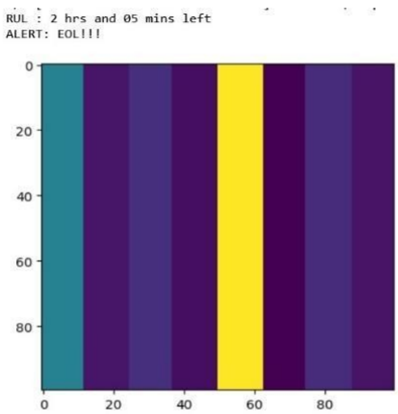

Fig 10c Predicted RUL for EOL condition

C. Histogram plots

The plot aims to visualize the distribution of weights within each layer of the BayesianCNN model. Analyzing weight distributions can provide insights into weight sparsity, magnitude, and potential issues of over fitting or underfitting. The x-axis weight values are the individual weight values associated with a specific parameter (Parameter 1 or Parameter 2) within a bayesian convolutional neural network layers. These values provide insights into the distribution of weights within each layer of the neural network, which can be helpful for understanding the learning dynamics of the model.

Parameter 1 refers to first set consisting of the kernel weights. These are the weights that are convolved with the input data to produce the output feature maps. Parameter 2 refer to different sets of weights associated with set consisting of the bias terms. Each filter typically has its own bias term. The density values on the histogram's y-axis represent the estimated probabilities of observing weight values in each bin on the x-axis. They are numerical representations of the likelihood of observing different weight values that aid in visualizing the weight distribution within each layer of the neural network.

Each of these images Fig 14a, 14b &14c represents the layers present in BCNN, the first layer being convolution layer, second is the dense layer 1 and lastly is the dense layer 2. For each of these layers, the weight distribution is interpreted. In the first layer, the weight values range from -0.2 to 0.1, indicating that the weights in this layer are relatively small in magnitude. This suggests that the model is designed to have a certain level of sparsity or regularization to prevent overfitting. The density of the weight distribution ranges from 0 to 5, so that it implies that the weights are somewhat concentrated around specific values, rather than being uniformly spread out. The bias values range from 0.001 to 0.1so that the model is designed to have a low bias effect. The density of the bias distribution ranges from 0 to 175, which is significantly higher than the weight density. This implies that there might be a larger number of unique bias values or that certain bias values are highly repetitive compared to the weights. These are the inferences observes from the first layer of BCNN. The small magnitude of weights and biases suggests that the model employs regularization techniques to prevent overfitting. Likewise, the inferences are observed for each of these layers.

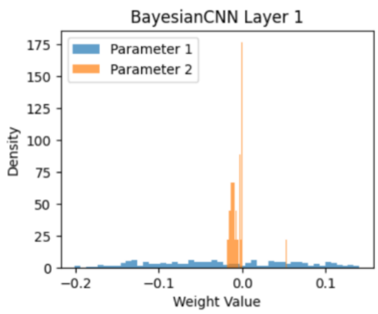

Fig 14a Visualizing weights of Convolutional Layer

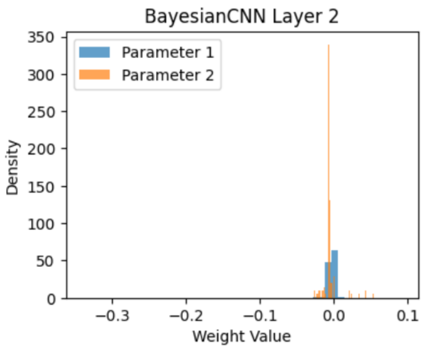

Fig 14b Visualizing weights of a Dense Layer 1

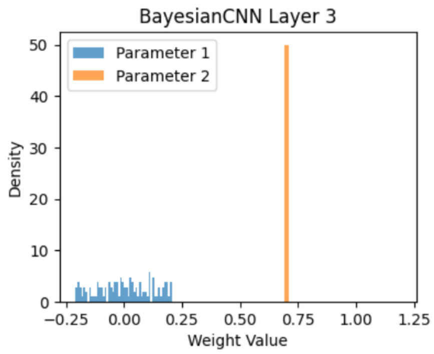

Fig 14c Visualizing weights of a Dense Layer 2

VII.CONCLUSION

Implementation of RUL estimation for LSS incorporates BCNN and transfer learning. This integration is crucial for ensuring mission reliability in submarines, where component functionality is vital for crew safety and operational success. BCNN models and fine-tuning strategies improve predictive accuracy and computational efficiency. The incorporation of Bayesian Dense Variational Inference layers enables uncertainty estimation, essential for risk assessment. Comparison with MSE based Stochastic CNN demonstrates BCNN's stability and performance, particularly with MAE as the loss function. Accurate estimation of RUL that is provided by the trained model in hours and minutes enhances overall dependability and contributes to mission success by facilitating real-time decision-making.

VIII. FUTURE ENHANCEMENTS

Firstly, integrating an emergency signal feature that activates in response to urgent situations, combined with real-time anomaly detection algorithms, could accelerate response times, ensure timely interventions in case of any emergencies. Secondly, developing a predictive maintenance dashboard that consolidates RUL predictions, system, and crew health metrics, could streamline decision-making processes for crew members. This user-friendly interface could notify the crew members with various human and ambient parameter levels by enabling them to make informed decisions about whether to continue with the mission or abort it. Thirdly, exploring advanced model architectures or ensemble methods may better capture complex relationships within the data, leading to improved predictive performance. Lastly, exploring the potential of integrating advanced sensor technologies, such as IoT-enabled devices, could provide more accurate data. This could offer deeper insights into the system's performance and degradation, further enhancing the precision and reliability of RUL predictions.

VIII.REFERENCES

[1] Victor Vantilborgh, Tom Lefebvre, Kerem Eryilmaz, & Guillaume Crevecoeur. (2023). Data-Driven Virtual Sensing for Probabilistic Condition Monitoring of Solenoid Valves. Ieee Transactions on Automation Science and Engineering, (99), 1–15. doi:10.1109/tase.2023.3287598

[2] Baoping Cai, Wen-Jun Li, Congkun Ren, Aibaibu Abulimiti, Xiaojie Tian, & Yanzhen Zhang. (2012). Probabilistic Thermal and Electromagnetic Analyses of Subsea Solenoid Valves for Subsea Blowout Preventers. Strojniški Vestnik, 58(11), 665–672. doi:10.5545/sv-jme.2012.681

[3] G. Mazaev, G. Crevecoeur and S. V. Hoecke, "Bayesian Convolutional Neural Networks for Remaining Useful Life Prognostics of Solenoid Valves With Uncertainty Estimations," in IEEE Transactions on Industrial Informatics, vol. 17, no. 12, pp. 8418-8428, Dec. 2021, doi: 10.1109/TII.2021.3078193.

[4] Kürşat İnce, Yakup Genc,Joint autoencoder-regressor deep neural network for remaining useful life prediction,Engineering Science and Technology, an International Journal,Volume 41,2023,101409,ISSN https://doi.org/10.1016/j.jestch.2023.101409.

[5] Yang, Jinsong & Peng, Yizhen & Xie, Jingsong & Wang, Pengxi. (2022). Remaining Useful Life Prediction Method for Bearings Based on LSTM with Uncertainty Quantification. Sensors. 22. 4549. 10.3390/s22124549.

[6] A Systematic Guide for Predicting Remaining Useful Life with Machine Learning.

[7] Daniel Camara Azevedo, Bernardete Ribeiro, & Alberto Cardoso. (2020). Prediction of the Remaining Useful Life of Aircraft Systems via Web Interface. International Journal of Online and Biomedical Engineering, 16, 23–32. doi:10.3991/ijoe.v16i04.11873

[8] Zhao-Hua Liu, Xudong Meng, Hua-Liang Wei, Long-Sheng Chen, Bi-Liang Lu, Zhenheng Wang, & Lei Chen. (2021). A Regularized LSTM Method for Predicting Remaining Useful Life of Rolling Bearings. International Journal of Automation and Computing, 18(4), 581–593. doi:10.1007/s11633-020-1276-6

[9] Fuqiang Sun, Xiang Li, Haitao Liao, & Xiankun Zhang. (2017). A Bayesian least-squares support vector machine method for predicting the remaining useful life of a microwave component. Advances in Mechanical Engineering, 9(1). doi:10.1177/1687814016685963

[10] Carroll James, Koukoura Sofia, McDonald Alasdair, Charalambous Anastasis, Weiss Stephan, & McArthur Stephen. (2019). Wind turbine gearbox failure and remaining useful life prediction using machine learning techniques. Wind Energy, 22(3), 360–375. doi:10.1002/we.2290

[11] Yuhuang Zheng. (2019). Predicting Remaining Useful Life Based on Hilbert Huang Entropy with Degradation Model. Journal of Electrical and Computer Engineering, 2019, 1–11. doi:10.1155/2019/3203959

[12] Joseph McGhee, Ian A. Henderson, Alistair Baird,Neural networks applied the and fault diagnosis of process valves and actuators,Measurement,Volume 20, Issue 4,1997,Pages 267-275,ISSN 02632241,https://doi.org/10.1016/S0263-2241(97)00043-2.

[13] G. Belforte, M. Velardocchia,Fault Detection and Dynamic Behaviour of Pneumatic Valves,IFAC Proceedings Volumes,Volume 27, Issue 5,1994,Pages 441446,ISSN 1474-6670,https://doi.org/10.1016/S1474-6670(17)48067-0.

[14] Lei, Wang & Cao, Hongrui & Ye, Zhisheng & Xu, Hao. (2023). Bayesian Large-kernel Attention Network for Bearing Remaining Useful Life Prediction and Uncertainty Quantification. Reliability Engineering & System Safety. 238. 109421. 10.1016/j.ress.2023.109421.

[15] G. Mazaev, G. Crevecoeur and S. V. Hoecke, Bayesian Convolutional Neural Networks for Remaining Useful Life Prognostics of Solenoid Valves With Uncertainty Estimations, in *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8418-8428, Dec. 2021, doi: 10.1109/TII.2021.3078193.

[16] Nagoor Basha Shaik, Srinivasa Rao Pedapati, Faizul Azly B A Dzubir,Remaining useful life prediction of a piping system using

artificial neural networks: A case study,Ain Shams Engineering Journal,Volume 13, Issue 2,2022,101535, ISSN 2090-4479, https://doi.org/10.1016/j.asej.2021.06.021.

[17] Yiwei Cheng, Chao Wang, Jun Wu, Haiping Zhu, C.K.M. Lee, Multi-dimensional recurrent neural network for remaining useful life prediction under variable operating conditions and multiple fault modes, Applied Soft Computing, Volume 118, 2022,108507,ISSN 1568-4946, https://doi.org/10.1016/j.asoc.2022.108507.

[18] Sheng Xiang, Yi Qin, Caichao Zhu, Yangyang Wang, Haizhou Chen,Long short-term memory neural network with weight amplification and its application into gear remaining useful life prediction,Engineering Applications of Artificial Intelligence,Volume 91,2020,103587, ISSN 0952-1976,https://doi.org/10.1016/j.engappai.2020.103587.

[19] Maximilian Benker, Lukas Furtner, Thomas Semm, Michael F. Zaeh,Utilizing uncertainty information in remaining useful life estimation via Bayesian neural networks and Hamiltonian Monte Carlo,Journal of Manufacturing Systems,Volume 61,2021,Pages 799-807,ISSN 0278-6125, https://doi.org/10.1016/j.jmsy.2020.11.005.

[20] Q. Yao, T. Yang, Z. Liu and Z. Zheng, "Remaining Useful Life Estimation by Empirical Mode Decomposition and Ensemble Deep Convolution Neural Networks," 2019 IEEE International Conference on Prognostics and Health Management (ICPHM), San Francisco, CA, USA, 2019, pp. 1-6, doi: 10.1109/ICPHM.2019.8819373.

[21] Cheng Zeng, Jinsong Huang, Hongrui Wang, Jiawei Xie, Yuting Zhang, Deep Bayesian survival analysis of rail useful lifetime,Engineering Structures,Volume 295,2023,116822,ISSN 0141-0296,https://doi.org/10.1016/j.engstruct.2023.116822.

[22] Rong Zhu, Yuan Chen, Weiwen Peng, Zhi-Sheng Ye,Bayesian deep-learning for RUL prediction: An active learning perspective,Reliability Engineering & System Safety,Volume 228,2022,108758,ISSN 0951-8320,https://doi.org/10.1016/j.ress.2022.108758.

[23] Donghui Pan, Jia-Bao Liu, Jinde Cao,Remaining useful life estimation using an inverse Gaussian degradation model,Neurocomputing, Volume 185,2016,Pages 64-72,ISSN 0925-2312,https://doi.org/10.1016/j.neucom.2015.12.041.

[24] Anil Kumar, Chander Parkash, Hesheng Tang, Jiawei Xiang,Intelligent framework for degradation monitoring, defect identification and estimation of remaining useful life (RUL) of bearing,Advanced Engineering Informatics,Volume 58,2023,102206,ISSN 1474-0346,https://doi.org/10.1016/j.aei.2023.102206.

[25] Niall M. O’Dowd, Ramin Madarshahian, Michael Siu Hey Leung, Joseph Corcoran, Michael D. Todd,A probabilistic estimation approach for the failure forecast method using Bayesian inference,International Journal of Fatigue,Volume 142,2021,105943,ISSN 0142-1123, https://doi.org/10.1016/j.ijfatigue.2020.105943.

[26] https://in.mathworks.com/help/predmaint/ug/rul-estimation-using-rul estimator-models.html

[27] https://englishprobabilistic-machine-learningelbo-interactive--or5u7m.streamlit.app/

[28] A Novel Transfer Learning Approach in Remaining Useful Life Prediction for Incomplete Dataset January 2022 IEEE Transactions on Instrumentation and Measurement 71:1-1 DOI:10.1109/TIM.2022.3162283

[29] https://in.mathworks.com/help/predmaint/ug/rul-estimation-using-rul

[30] https://mpatacchiola.github.io/blog/2021/01/25/intro-variational-inference.html

[31] Bayesian Neural Networks: An Introduction and Survey 22 Jun 2020 https://doi.org/10.1007/978-3-030-42553-1_3

[32] Variational encoding approach for interpretable assessment of remaining useful life estimation Volume 222, June 2022, 108353 Reliability Engineering & System Safety https://doi.org/10.1016/j.ress.2022.108353

[33] https://towardsdatascience.com/understanding-the-3-most-common-loss-functions-for-machine-learning-regression-23e0ef3e14d3