Self-supervised Approach to Degradation Trajectory Retrieval and Diffusion-based Failure Generation

Anonymous submission

Abstract

The anomaly generation in system monitoring plays a crucial role in training and testing methodologies for deep learning-based state control. However, existing methods primarily focus on generating isolated anomaly samples for state monitoring. We propose a new approach to generating trajectories of system degradation using a conditional diffusion model with a self-supervised method to extract interpretable degradation patterns as the conditioning mechanism. We develop a complex loss function based on metric learning, regression, and reconstruction losses to construct a low-dimensional and interpretable latent space. We used an LSTM-based autoencoder using the proposed loss. We demonstrate that the model achieves interpretable fault modes and accurately divides associated degradation trajectories through unsupervised clustering on the CMAPSS aircraft engine dataset. We employed extrapolation to generate the new data based on the most similar degradation pattern and trajectory using latent spatial properties. That latent representations served as conditional inputs for a generative diffusion model based on an adapted U-Net architecture tailored for time series data. We evaluate the quality of the latent space in the autoencoder by assessing its accuracy in estimating healthy states and using the original evaluation metric of the CMAPSS test dataset achieving close to state-of-the-art results. Additionally, we test the generative model by comparing the generated degradation trajectories with time series data captured near the points of failure. The proposed approach offers a novel and effective solution for self-supervised complete trajectories generation with more than 10% efficiency than the counterparts and a deeper understanding of fault modes in complex systems. We demonstrate the method's effectiveness and practical applicability through the model evaluation on the open C-MAPSS dataset tasks.

Introduction

The increasing complexity of systems determines the growth in monitoring techniques to detect anomalies, deviations from a regular operation, and potential failures (Zhao et al. 2019). The anomaly generation for time-series-based system monitoring applies to developing methods for classical ML techniques and neural network training data augmentation (Talavera et al. 2022). Existing anomaly generation methods focus on the augmentation or generation of isolated anomaly samples for state monitoring. At the same time, system behaviour is often dynamic, and complete degradation patterns are of great interest.

The lack of techniques capable of generating complete degradation trajectories poses several challenges. Firstly, the availability of full trajectories in training enables more accurate system state predictions, empowering proactive maintenance actions. Secondly, the limited understanding of underlying patterns and causes of system degradation hampers the development of effective monitoring and fault diagnosis strategies. Finally, comprehensive degradation data is essential for evaluating and comparing monitoring algorithms and techniques.

In this paper, we propose a new method for generating trajectories of system degradation. Our idea is closely related to the approach to video series processing with self-supervised extraction of degradation trajectories in the latent space and using the obtained patterns to generate conditional inputs for a generative diffusion model. We developed a novel complex loss function based on trajectory-based metric learning, regression, and reconstruction losses to construct a low-dimensional and interpretable latent space. Using the acquired loss, we trained a modified low-dimensional LSTM-based TSHAE (time-series hybrid autoencoder). We demonstrated that the model achieves interpretable fault modes and accurately divides associated degradation trajectories through unsupervised clustering. The division of the latent space aligned perfectly with the ground truth of the dataset. We employed extrapolation based on the most similar degradation pattern and trajectory using latent spatial properties to generate the initial test degradation trajectory extension.

The latent representations obtained from the autoencoder served as conditional inputs for a generative diffusion model based on an adapted U-Net architecture (Ronneberger, Fischer, and Brox 2015) tailored for time series data. We evaluated the quality of the latent space in the autoencoder by assessing its accuracy in estimating healthy states and using the original evaluation metric of the C-MAPSS (Saxena et al. 2008) test dataset to achieve close to state-of-the-art results. The C-MAPSS dataset is widespread, and because of this, the proposed approach is easily comparable and reproducible. We also tested the generative model by comparing the generated degradation trajectories with time series data captured near the points of failure.

The main contributions of this work are as follows:

- We proposed to use the low-dimensional, high-regularized autoencoder trained with hybrid degradation trajectory loss function to research the underlying data structure.
- We have performed the testing of loss based on the trajectory-base triplet, RUL RMSE, KL-divergence, and reconstruction losses to generate interpretable two-dimensional latent space and obtained close-to-SOTA results for the NASA C-MAPSS problem with TSHAE (time-series hybrid autoencoder) based on the LSTM-based architecture.
- We developed a new method of using relevant degradation trajectories in latent space as conditional input sources for the diffusion model to generate new or continuation of existing degradation trajectories near system failure.
- We confirmed the method's performance for generating plausible trajectories on the NASA C-MAPSS dataset by developing trajectory extensions using the diffusion model with conditions based on the projection in latent space of the nearest trajectory in the degradation pattern.

The rest of the paper is organized as follows. The section "Related work" briefly gives the related research to the approaches to time-series augmentation, latent space research, and system RUL estimation. The section "Proposed approach and methods" describes the proposed generative process and used model structures. The section "Dataset" briefly gives the C-MAPSS dataset review and proposed method adaptation. The section "Experiments and Discussion" includes a description of the learning process, experiments on the proposed models, and their performance study. The section "Conclusion" summarizes our research and includes possible feature work. Finally, the "Reproducibility" gives system information for experiment reproducibility.

Related work

In recent research, several time-series augmentation methods have been proposed to enhance the performance of various tasks. Tu et al. (Tu et al. 2018) introduce LSTM-AE. This novel LSTM autoencoder network combines LSTM and an autoencoder architecture for improved spatial-temporal data augmentation in skeleton-based human action recognition. Hatamian et al. (Hatamian et al. 2020) investigate the effectiveness of various data augmentation methods, such as GANs and GMMs, in addressing class imbalance for atrial fibrillation (AF) classification from ECG signals, demonstrating improved accuracy and f1-score compared to traditional oversampling techniques. Zhu et al. (Zhu et al. 2019) propose BiLSTM-CNN GAN, a generative adversarial network for synthesizing ECG data that closely resembles accurate clinical records, addressing the challenge of acquiring labelled data for automated medical-aided diagnosis. Lou et al. (Lou, Qi, and Li 2018) introduce a supervised signal-based approach in Wasserstein GAN for one-dimensional data augmentation, achieving improved convergence rate and sample quality compared to traditional GAN variants. Wang et al. (Wang, Kim, and Lee 2019) propose a voice conversion approach using WaveNet for speech augmentation in automatic speech recognition, achieving better generalization than traditional methods and further improvement when combined with speed perturbation. Cao et al. (Cao, Tan, and Pang 2014) propose a parsimonious mixture of Gaussian trees model for oversampling in imbalanced and multimodal time-series classification, demonstrating improved classification accuracy compared to state-of-the-art methods. Lastly, Kang et al. (Kang, Hyndman, and Li 2020) propose GRATIS, a time series generation method based on a mixture of autoregressive models, for simulating diverse and controllable benchmarking data in time series analysis tasks. These advancements in data augmentation methods contribute to improving various Supplementary materials in different domains.

Several works have explored interpretable latent representations and disentangled learning. Costa et al. (Costa and Sánchez 2022) reduced dimensionality in the latent space of a modified VAE model, while EEG (Wagh et al. 2022) examined the association between input signal shifts and latent space disentanglement. ChromaVAE (Yang et al. 2022b) addressed the preference for encoding shortcuts, while LostLS (Montero et al. 2022) measured disentanglement using the DCI metric. PrototVAE (Gautam et al. 2022) learned prototypes and a classifier, and RotVAE (Nasiri and Bepler 2022) separated latent representations into rotation, translation, and semantics.

The RUL prediction methods can be categorized into direct prediction without auxiliary features (Da Xu 2020; Lomov et al. 2021), optimizing linear RUL curves (Jayasinghe et al. 2019), using a constant area of the RUL curve (Jayasinghe et al. 2019; Berghout et al. 2020), and employing feature extractors with regression heads (Gebraeel et al. 2004; Zheng et al. 2017; Wu et al. 2018; Peng et al. 2020; Hong et al. 2020; Zhu, Chen, and Peng 2018; Zhao et al. 2017).

Proposed approach and methods

In this paper, we developed a new Self-supervised Pipeline for generating time-series degradation trajectories up to system failure. The pipeline includes a hybrid approach to self-supervised-based latent space generation and unsupervised clustering of degradation patterns in training dataset latent space. The second part of the pipeline includes a condition-based diffusion model to generate plausible degradation trajectories using latent representations of degradation trajectories as conditional inputs.

System degradation patterns In the general case, the evolution of the state of a complex system is not limited to a single process. Under external conditions, it is difficult to distinguish a trajectory specific to a particular degradation pattern at the initial operational time, and such a state is called a healthy state. However, the possibility of singling out specific interpretable trajectories of system degradation appears when going outside of a healthy state.

In this paper, we consider physical systems, which change according to the continuous change of physical parameters described by a system of differential equations. Interpolation of failure patterns has an explicit physical meaning in the form of observable trajectories of processes. The number of trajectories is limited, and the system cannot degrade to failure without matching some pattern associated with the degradation of a system element. Thus, it is possible to consider device degradation patterns as manifold approximations (Goodfellow, Bengio, and Courville 2016), and degradation trajectories at the initial moment represent the result of interpolation between them, which later converges to one of them (i.e., to a particular system failure). So, we compose a Loss function to map the data into degradation failure-related manifold representation. In that case, we can ensure that the latent space shift of unseen data would be close to the characteristic degradation pattern, which is the focus of this paper.

Degradation patterns approximation The most natural way to investigate the underlying data structure is to use autoencoders (Zemouri et al. 2022).

The TSHAE encoder Fig. 1 comprises a single unidirectional LSTM layer with a hidden size of 300. Following the LSTM layer, the outputs are directed through two identical blocks, consisting of a Dropout layer with a dropout rate of 0.2 and a Fully Connected layer with 2 neurons. These blocks are utilized for estimating the values of μ and $\log \sigma^2$. The resulting output from the TSHAE encoder is a latent representation $\boldsymbol{z} \in \mathbb{R}^2$, which is sampled from a Gaussian distribution parameterized by the estimated values of μ and σ^2. The TSHAE decoder comprises a unidirectional LSTM layer with a hidden size of 300, followed by a Dropout layer with a dropout rate of 0.8, and a second unidirectional LSTM layer with a hidden size equal to the input data feature dimensions. The additional regressor head contains a Fully Connected layer with 100 neurons with a Tanh activation function, followed by a Dropout layer with a dropout rate of 0.5 and a Fully Connected layer with a single neuron.

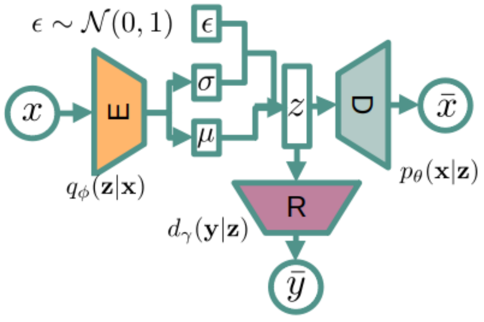
$\epsilon \sim \mathcal{N}(0,1)$
ϵ
σ
μ
x
E
$q_\phi(\mathbf{z}|\mathbf{x})$
z
D
$\bar{x}$
$p_\theta(\mathbf{x}|\mathbf{z})$
R
$d_\gamma(\mathbf{y}|\mathbf{z})$
$\bar{y}$

Figure 1: TSHAE architecture for encoding degradation trajectories. We preserved it simple and direct to illustrate the effectiveness of the approach for the C-MAPSS problem.

Training The trajectory-based loss function part aims to effectively model the stretching of trajectories by separating healthy and unhealthy states while also grouping closely related states, particularly those indistinguishable from healthy states. It achieves this through a combination of the Triplet Loss and the KL Divergence:

$$\mathcal{L}_{\text{trajectory}} = \lambda \sum_{i=1}^{N_p} \max\left(0, d(A_i, P_i) - d(A_i, N_i) + \alpha\right) + \\ + \sum_{i=1}^{N} q_\phi(z_i|x_i) \log\left(\frac{q_\phi(z_i|x_i)}{p_\theta(z_i)}\right) \tag{1}$$

where N_p represents the total number of triplets in the training set, A_i, P_i, and N_i denote the embeddings of the anchor, positive, and negative samples in the ith triplet, respectively, $d(\cdot,\cdot)$ represents a Euclidean distance metric, α is a margin that controls the separation between positive and negative samples, $q_\phi(z|x)$ is learned approximate posterior distribution over latent variables z given input data x, $p_\theta(z)$ is the prior distribution over latent variable z, N is total number of embedding in the training set λ represents the weight assigned to the Triplet Loss term, and its value is set to 150.

The space organization-based loss function includes the use of the classic reconstruction and RUL MSE losses:

$$\mathcal{L}_{\text{space}} = \frac{1}{N_t} \sum_{i=1}^{N_t} (x_i - \hat{x}_i)^2 + \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \tag{2}$$

where X represents the original signal or image, $\hat{X}$ represents the reconstructed signal or image, and N_t represents the total number of time-series points, Y represents the true RUL values, $\hat{Y}$ represents the predicted values, N represents the total number of predicted values, y_i denotes the true RUL value for the ith sample, and $\hat{y}_i$ represents the predicted RUL value for the same sample.

Despite using triplet loss having no significant effect on C-MAPSS RUL results (the Application section), separating healthy and unhealthy device states within the same trajectory improves semantic separability, which is important in real-world tasks. For the proposed method, we expected the step-by-step trajectory generation improvement in the presence of classical real-world train data issues: high noise, missing values and incorrect labelling.

The TSHAE was trained using the Adam optimizer, with a constant learning rate of 0.002 and zero weight decay, for 30 epochs. To find the better parameters, we use grid search on parameters: TSHAE encoder number of LSTM layers, encoder LSTM number of units, encoder dropout rate, TSHAE regression head number of neurons, and regression head dropout rate.

Failure trajectory latent generation

Because of the dataset's small amount, we cannot build a plausible representation of the whole latent space. So, assuming that system behaviour converges to a set of particular failures, we proposed that close failure trajectories may join into groups in the latent space. We have used unsupervised clustering of trajectories near failures to isolate the trajectories that led to a given failure. We consider a two-dimensional trajectory representation a time series and perform K-means clustering based on the Euclidean distance. The approach is presented in Algorithm 1.

```
Algorithm 1: Trajectory generation based on latent degradation pattern clustering
Input: latent degradation trajectories from the train dataset; new sensor value trajectory beginning; latent-based condition diffusion model.
Output: generated time series of device sensor readings up to failure.
  1: Translate trajectory to latent space: translate sensor value trajectory into the latent space using the trained encoder of the proposed model.
  2: Find the closest cluster: determine whether the last points of the trajectory belong to each class in the latent space by calculating the distance to the cluster elements.
  3: Find the closest trajectory: find the closest trajectories in the closest cluster by Dynamic Time Warping similarity measure.
  4: Approximate difference with quadratic spline: approximate the difference between the initial trajectory and weighted three closest trajectories using a quadratic spline.
  5: Transfer behavior by extrapolating the spline: extrapolate the spline to transfer the behavior from the cluster, ensuring common latent space shift within the same failure pattern.
  6: Generate trajectory leading to failure: use the resulting trajectory points as conditions for the diffusion model.
```

Failure latent for conditional diffusion-based generation An Information bottleneck can be helpful for information extraction and improving interpretability when describing the internal structure. However, from the point of view of decoder-based generation, it creates severe limitations on the quality of the reconstructed series. We use a class of actively evolving diffusion models as a generative model and use the data obtained from the analysis of internal structure as conditional input. Using diffusion models by analogy with pictures allows us to solve a wide range of problems (Yang et al. 2022a) - from the generation of degradation trajectories and failures of complex systems to the interpolation of missing values of controlling sensors. The model's architecture is shown in Fig. 2 and is based on the popular U-Net, adapted for use in time series. We used a basic architecture with minor changes to keep the experiment clean and reproducible. The model's conditioning method follows a similar approach to the one presented in (Ho and Salimans 2021). The model incorporates timestep embeddings t_e and context embeddings c_e into the U-Net activations at a particular layer a_L using the following equation: $a_{L+1} = c_e a_L + t_e$.

Denoising Diffusion Probabilistic Model (DDPM) Loss function is defined as Mean Squared Error between the noise added in the forward process and the noise predicted by the model:

$$\mathcal{L}_{t,c} = \mathbb{E}_{\mathbf{x}_0,t,\epsilon,c}\left[||\epsilon - \epsilon_\theta\left(\sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, c\right)||^2\right] \tag{3}$$

Where $\mathbf{x}_0 \in \mathbb{R}^{M\times N}$ is input time series data point, α_t and $\bar{\alpha}_t$ are defined as $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=0}^{t} \alpha_s$ and β_t is variance at time step t. The function ϵ_θ is neural network approximation of noise ϵ and c is conditioning variable.

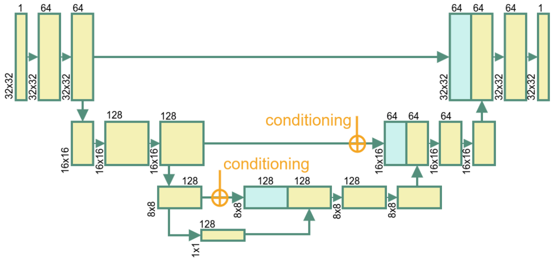

Figure 2: Adapted for time-series U-Net (Ronneberger, Fischer, and Brox 2015) based conditional model

Training DDPM was trained on the train portion of the CMAPSS fd003 subdataset. To address the imbalance in the target variable distribution, samples with RUL ≥ 125 were replaced by those with RUL < 40 with a probability of 0.9. This approach helped to create a more representative training set. DDPM parameters: number of noise steps $T = 500$, $\beta_0 = 10^{-4}$, $\beta_T = 0.02$. The Diffusion model was optimized using the Adam optimizer for 100 epochs. A linear learning rate scheduler was utilized, where the learning rate lr at each epoch was determined by the function $lr(epoch) = 3 \times 10^{-3} \times \left(1 - \dfrac{epoch}{100}\right)$.

Dataset

The dataset (Saxena et al. 2008) encompasses the data generated by the ”Commercial Modular Aero-Propulsion System Simulation” (C-MAPSS). This simulation program was used to emulate the behaviour of turbofan engine sensors until failure, encompassing a wide range of flight conditions. The C-MAPSS dataset comprises four distinct subsets of data, each containing readings from a set of 21 sensors strategically positioned throughout different components of the degrading engine. These sensors are influenced by three crucial operational settings: height, speed, and acceleration.

The simulation was employed to investigate the behaviour of sensors as the engine gradually deteriorated, leading to failure. In order to approximate the complex real-world engine behaviour, various types of noise were incorporated into the modelled engine sensor readings (Saxena et al. 2008). The training dataset was divided into training and validation subsets, with a train fraction of 0.8 and a fixed random seed of 42. Condition-based standardization was applied, individually standardizing the data for each operational setting. A downsampling strategy was used during validation to balance the highly imbalanced target distribution in the training dataset. Samples with RUL ≥ 125 were replaced by those with RUL < 125 with a probability of 0.8.

RUL problem statement

Target The RUL problem is the basic formulation of the C-MAPSS dataset problem, where the Remain useful life characterizes the remaining length of the degradation trajectory.

Because of the weak distinguishability of the healthy state, a piece-wise RUL is used as the target function. The state is considered healthy above the number RUL equal to the specific value V, and the regression model should return N as RUL prediction:

$$RUL = \begin{cases} V, & \text{if } RUL \geq V \\ RUL, & \text{if } RUL < V. \end{cases} \tag{4}$$

Where V is an heuristic parameter, and in most cases for the C-MAPSS dataset task $V = 125$ value is used due to previously performed empirical study (Heimes 2008).

Metrics The performance of the models on the C-MAPSS dataset RUL problem was evaluated using RMSE and Score metric on the original test dataset. RMSE metric definition was discussed before, and Score metric is:

$$\text{Score} = \sum_{i=1}^{N} \begin{cases} e^{-\frac{\hat{y}_i - y_i}{13}} - 1, & \text{if } \hat{y}_i - y_i < 0 \\ e^{\frac{\hat{y}_i - y_i}{10}} - 1, & \text{if } \hat{y}_i - y_i \geq 0 \end{cases} \tag{5}$$

where $\hat{y}_i$ and y_i are predicted and true value of RUL for engine i, N is number of test samples. The asymmetric Score metric presented in the original paper for PHM2008 competition penalizes overestimations in predictions due to maintenance regulations.

Up to failure trajectory generation problem statement

Target We evaluate trajectory generation quality on the validation dataset due to limited low RUL values in the test data. We aim to generate plausible sensor behaviour that closely resembles the original readings. The generated trajectories from the diffusion model should align with the original degradation pattern after re-passing through the encoder. Failure trajectory generation is a reconstruction of the trajectory from 72 cycles of device operation of sensor readings before failure. Classification of trajectories in the parameter space was performed on the remaining 53 points, in which the degradation presents according to the piece-wise RUL statement (before the 125 RUL value). However, this is not a time-series forecasting task while we generate a new plausible degradation trajectory rather than reconstructing an existing one.

Metrics The accuracy of belonging of the generated part of the trajectory to the same pattern as the original data was solved in the binary classification statement using the accuracy metric:

$$Accuracy = \frac{TP}{TP + FN} \tag{6}$$

where TP represents the number of True Positives (correctly predicted degradation pattern class), FN represents the number of False Negatives (incorrectly predicted degradation pattern class).

To compare the generated time series, we used the DTW metric in order to rely on the matching shapes of the trajectories qualitatively. The DTW distance between two time-series, denoted as $\text{DTW}(X, Y)$, is computed using the following equation:

$$\text{DTW}(X, Y) = \min_{\text{alignment}} \left(\sum_{i=1}^{N} \sum_{j=1}^{M} c(i, j) \right) \tag{7}$$

where X and Y represent the two time-series being compared, and N and M represent their respective lengths. The cost function $c(i, j)$ quantifies the dissimilarity between the ith point in time-series X and the jth point in time-series Y by Euclidean metric, $\min_{\text{alignment}}$ indicates that the metric is calculated for the optimal relative to alignment of the time-series points between X and Y.

Name (year)	RMSE, Score
TCNN-Transformer (2021)	11.35, 415
RVE (2022)	12.51, 256
BiGRU-TSAM (2022)	12.45, 232
ADL-DNN (2022)	12.59, 281
KGHM (2023)	13.64, 333
TSHAE, ours (2023)	**10.95**, **189**

Table 1: Test results for fd003 C-MAPSS subdataset

Experiments and Discussions

The first task performed was to select the architecture and training conditions of the model in order to optimize the result on validation by two primary metrics: RMSE and the error of location of degrading states in healthy regions. Next, for illustration, the article will look at sub-dataset number 3; figures and related metrics for the sub-datasets are presented in the Supplementary materials. The optimal model's performance metrics on the test subset of the dataset for the RUL task are presented in Table 1; the average and range resulting from 10 different seed variations under constant parameters are as follows: an average RMSE of 12.99 ± 1 and an average score of 444 ± 225. The models for comparison in Table 1 were chosen based on the date of the work and the relative model complexity.

In terms of an auxiliary ablation study, the problem of the effect of extraneous noise on the latent space and the predicted RUL value makes sense within the problem statement. We performed simulations on Fig. 3, varying the noise from 0 to 0.5. There is practically no effect on RUL since the latent space of dimension 2 plays the role of a noise modifier, but the noise significantly degrades the Healthy unhealthy metric. Extended simulation results are presented in the Supplementary materials.

The latent state image shows two groups associated with the engine fault modes mentioned in the accompanying dataset documentation. We clustered the last 20 points assuming that a particular failure corresponds to a particular degradation pattern. The clustering result is shown in Fig. The clustering result is shown in Fig. 4. As a result of the time-series analysis, we determined that the time-series gauge with numbers 7, 9, and 14 best illustrated the differences in the trajectories on the time-series. Further, we used the readings of these sensors and the result of the mapping

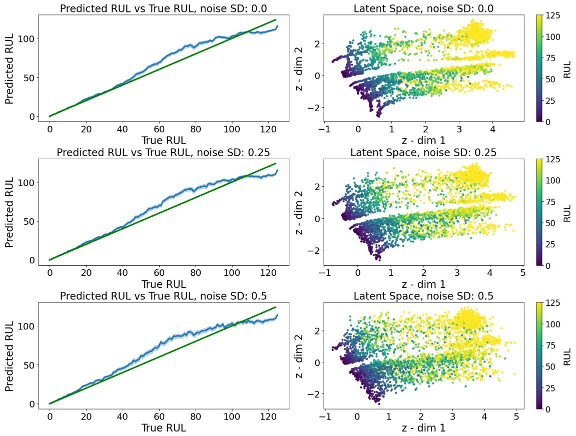

Figure 3: Latent space and inverted RUL prediction robustness relative to noisy input. The resulting metric values were 0.0385 for a noise standard deviation (SD) of 0, 0.0505 for SD 0.25, and 0.0592 for SD 0.5.

to the latent space to illustrate the quality of the generative model. The sensor-based cluster representations are in Supplementary materials.

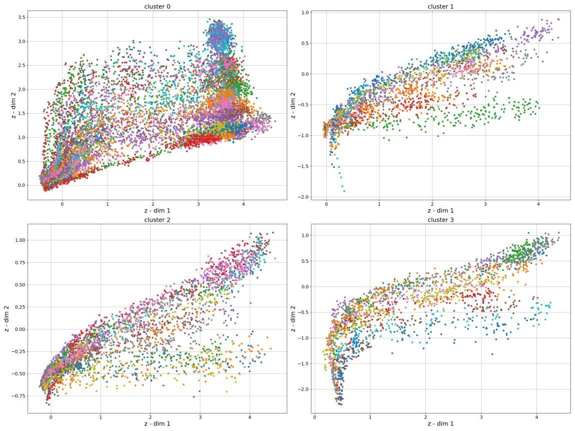

Figure 4: Clusters of C-MAPSS fd003 sub-dataset latent degradation trajectories. Each point in latent space represents a specific engine state relative to RUL and reconstruction information.

Within the selected clusters, there is a high variation of trajectories, which, due to their small number (80), does not allow us to identify the degradation trajectory pattern correctly with a small margin of error. However, we proposed that near the failure region, the device degradation trajectory should change according to the same pattern with the accuracy of the influence of other degradation processes. It is important to note that using KL divergence as one of the loss functions also significantly affects the appearance of pointwise deviation of the trajectories. Let us perform approximation and extrapolation of the data in the latent space according to Algorithm 1. The result is shown in Fig. 5.

Next, we performed training of the U-Net-based diffusion model using the time series of the trane and selected latent states as a condition. The results of the generation are shown in Fig. In particular, all the generations with a given latent from a particular cluster can generate a trajectory belonging to that cluster during the generation problem. Since when training the diffusion model, we did not put an auxiliary loss on the conservation of those features on which the regression problem is solved, the predictions of the lifetime of the generated trajectories are noisier. The example is in Fig. 6.

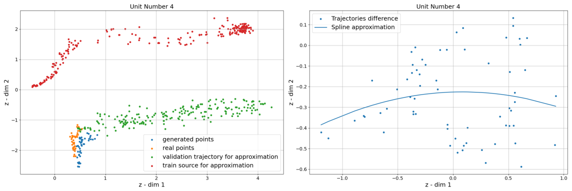

Figure 5: Validation trajectory extrapolation example using the difference between weighted closest train trajectories and the validation trajectory beginning associated with the cluster 1.

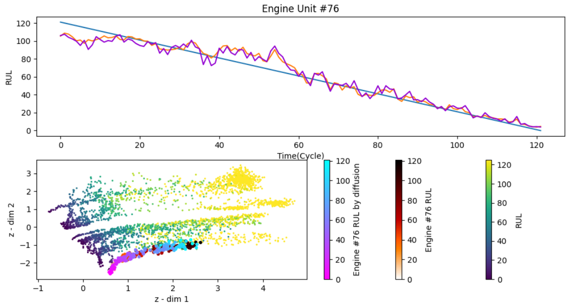

Figure 6: Diffusion model generation with latent representation condition built from real validation data encoding

Then we used the result of approximation in latent space to generate continuation trajectories on validation. Running inference for a basic DDPM model consumes 7.5 GB of VRAM and requires approximately 22 minutes on NVIDIA RTX A4000 GPU. The results are shown in Fig. 7, and the results of generating other engines of the validation dataset are presented in Supplementary.

The main goal of the research was to develop a method that allows for generating a plausible degradation trajectory near failure. The generation results demonstrated the exact character 19 from 20 generated trajectories due to conditional extrapolation for the diffusion model. Similarly, for the validation data used to control the quality of the sensor approximation, a qualitative match was obtained for most of the trajectories.

To the best of our knowledge, generating a relevant conditional vector that can be used to generate a time series is not a competitive task, so comparisons with other models are limited. Quantitative metrics values were used to select the best diffusion model checkpoint. On the validation dataset for fd003, RMSE is 0.51, and DTW is 26.59. Other sub-

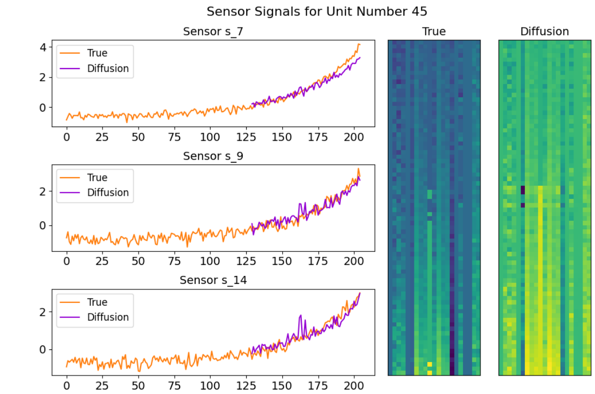

Figure 7: Diffusion model generation with latent extrapolation points as a condition

dataset results are presented in Supplementary materials.

Conclusion

This paper proposes a novel approach for investigating the underlying data structure using a low-dimensional, high-regularized autoencoder trained with a hybrid degradation trajectory loss function. We successfully generated an interpretable two-dimensional latent space by training with complex metrics, including trajectory-based triplet loss, remaining useful life (RUL) root mean square error (RMSE), KL divergence, and reconstruction losses. Furthermore, our method based on the time-series hybrid autoencoder (TSHAE) achieved results close to the state-of-the-art (SOTA) performance on the NASA C-MAPSS problem. We developed a new method of using relevant degradation trajectories in latent space as conditional inputs for the diffusion model to generate new or continuation of existing degradation trajectories near system failure. We confirmed the method's performance for generating plausible trajectories on the NASA C-MAPSS dataset by generating trajectory extensions using the diffusion model with conditions based on the projection in latent space of the nearest trajectory in the degradation pattern. For the C-MAPSS dataset, the proposed solution and the loss function demonstrated sufficient results for latent space-based RUL prediction and trajectory generation. Implemented GMAE (Kingma et al. 2014; Dilokthanakul et al. 2017) shown in the Application did not significantly improve the results, but we recommend using more advanced models for real-world problems. Further work on developing the approach could improve the loss function to achieve better degradation pattern representation using metric learning and graph representations.

Reproducibility

OS: Ubuntu 22.04.3 LTS GPU: Nvidia RTX A4000 16 Gb CPU: Intel® Core™ i5-12600K(F) To reproduce the result use following libraries versions: python 3.9.16; numpy 1.24.3;pandas 1.5.3; matplotlib 3.7.1; torch 2.0.1; scikit learn 1.2.0; scipy 1.9.3; tqdm 4.65.0; tslearn 0.6.1; PyTorch seed is 42, numpy seed is 42, tslearn random_state is 42, PyTorch deterministic algorithms are on.

References

Berghout, T.; Mouss, L.-H.; Kadri, O.; Saïdi, L.; and Benbouzid, M. 2020. Aircraft engines remaining useful life prediction with an improved online sequential extreme learning machine. *Applied Sciences*, 10(3): 1062.

Cao, H.; Tan, V. Y.; and Pang, J. Z. 2014. A parsimonious mixture of Gaussian trees model for oversampling in imbalanced and multimodal time-series classification. *IEEE transactions on neural networks and learning systems*, 25(12): 2226–2239.

Costa, N.; and Sánchez, L. 2022. Variational encoding approach for interpretable assessment of remaining useful life estimation. *Reliability Engineering and System Safety*, 222: 108353.

Da Xu, L. 2020. Industrial information integration–An emerging subject in industrialization and informatization process.

Dilokthanakul, N.; Mediano, P. A. M.; Garnelo, M.; Lee, M. C. H.; Salimbeni, H.; Arulkumaran, K.; and Shanahan, M. 2017. Deep Unsupervised Clustering with Gaussian Mixture Variational Autoencoders. arXiv:1611.02648.

Gautam, S.; Boubekki, A.; Hansen, S.; Salahuddin, S. A.; Jenssen, R.; Höhne, M. M.; and Kampffmeyer, M. 2022. ProtoVAE: A Trustworthy Self-Explainable Prototypical Variational Model.

Gebraeel, N.; Lawley, M.; Liu, R.; and Parmeshwaran, V. 2004. Residual life predictions from vibration-based degradation signals: a neural network approach. *IEEE Transactions on industrial electronics*, 51(3): 694–700.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Hatamian, F. N.; Ravikumar, N.; Vesal, S.; Kemeth, F. P.; Struck, M.; and Maier, A. 2020. The effect of data augmentation on classification of atrial fibrillation in short single-lead ECG signals using deep neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1264–1268. IEEE.

Heimes, F. O. 2008. Recurrent neural networks for remaining useful life estimation. *2008 International Conference on Prognostics and Health Management*, 1–6.

Ho, J.; and Salimans, T. 2021. Classifier-Free Diffusion Guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.

Hong, C. W.; Lee, C.; Lee, K.; Ko, M.-S.; Kim, D. E.; and Hur, K. 2020. Remaining useful life prognosis for turbofan engine using explainable deep neural networks with dimensionality reduction. *Sensors*, 20(22): 6626.

Jayasinghe, L.; Samarasinghe, T.; Yuenv, C.; Low, J. C. N.; and Ge, S. S. 2019. Temporal convolutional memory networks for remaining useful life estimation of industrial machinery. In *2019 IEEE International Conference on Industrial Technology (ICIT)*, 915–920. IEEE.

Kang, Y.; Hyndman, R. J.; and Li, F. 2020. GRATIS: GeneRAting TIme Series with diverse and controllable characteristics. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 13(4): 354–376.

Kingma, D. P.; Rezende, D. J.; Mohamed, S.; and Welling, M. 2014. Semi-Supervised Learning with Deep Generative Models. arXiv:1406.5298.

Lomov, I.; Lyubimov, M.; Makarov, I.; and Zhukov, L. E. 2021. Fault detection in Tennessee Eastman process with temporal deep learning models. *Journal of Industrial Information Integration*, 23: 100216.

Lou, H.; Qi, Z.; and Li, J. 2018. One-dimensional data augmentation using a Wasserstein generative adversarial network with supervised signal. In *2018 Chinese Control And Decision Conference (CCDC)*, 1896–1901. IEEE.

Montero, M. L.; Bowers, J. S.; Costa, R. P.; Ludwig, C. J. H.; and Malhotra, G. 2022. Lost in Latent Space: Disentangled Models and the Challenge of Combinatorial Generalisation.

Nasiri, A.; and Bepler, T. 2022. Unsupervised Object Representation Learning using Translation and Rotation Group Equivariant VAE.

Peng, J.; Wang, S.; Gao, D.; Zhang, X.; Chen, B.; Cheng, Y.; Yang, Y.; Yu, W.; and Huang, Z. 2020. A hybrid degradation modeling and prognostic method for the multi-modal system. *Applied Sciences*, 10(4): 1378.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv:1505.04597.

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

Talavera, E.; Iglesias, G.; González-Prieto, Á.; Mozo, A.; and Gómez-Canaval, S. 2022. Data augmentation techniques in time series domain: a survey and taxonomy. *arXiv preprint arXiv:2206.13508*.

Tu, J.; Liu, H.; Meng, F.; Liu, M.; and Ding, R. 2018. Spatial-temporal data augmentation based on LSTM autoencoder network for skeleton-based human action recognition. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, 3478–3482. IEEE.

Wagh, N.; Wei, J.; Rawal, S.; Berry, B. M.; and Varatharajah, Y. 2022. Evaluating Latent Space Robustness and Uncertainty of EEG-ML Models under Realistic Distribution Shifts.

Wang, J.; Kim, S.; and Lee, Y. 2019. Speech augmentation using wavenet in speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 6770–6774. IEEE.

Wu, Y.; Yuan, M.; Dong, S.; Lin, L.; and Liu, Y. 2018. Remaining useful life estimation of engineered systems using vanilla LSTM neural networks. *Neurocomputing*, 275: 167–179.

Yang, L.; Zhang, Z.; Song, Y.; Hong, S.; Xu, R.; Zhao, Y.; Shao, Y.; Zhang, W.; Cui, B.; and Yang, M.-H. 2022a. Diffusion models: A comprehensive survey of methods and applications. *arXiv preprint arXiv:2209.00796*.

Yang, W.; Kirichenko, P.; Goldblum, M.; and Wilson, A. G. 2022b. Chroma-VAE: Mitigating Shortcut Learning with Generative Classifiers.

Zemouri, R.; Lévesque, M.; Boucher, É.; Kirouac, M.; Lafleur, F.; Bernier, S.; and Merkhouf, A. 2022. Recent research and applications in variational autoencoders for industrial prognosis and health management: A survey. In *2022 Prognostics and Health Management Conference (PHM-2022 London)*, 193–203. IEEE.

Zhao, R.; Wang, D.; Yan, R.; Mao, K.; Shen, F.; and Wang, J. 2017. Machine health monitoring using local feature-based gated recurrent unit networks. *IEEE Transactions on Industrial Electronics*, 65(2): 1539–1548.

Zhao, R.; Yan, R.; Chen, Z.; Mao, K.; Wang, P.; and Gao, R. X. 2019. Deep learning and its applications to machine health monitoring. *Mechanical Systems and Signal Processing*, 115: 213–237.

Zheng, S.; Ristovski, K.; Farahat, A.; and Gupta, C. 2017. Long short-term memory network for remaining useful life estimation. In *2017 IEEE international conference on prognostics and health management (ICPHM)*, 88–95. IEEE.

Zhu, F.; Ye, F.; Fu, Y.; Liu, Q.; and Shen, B. 2019. Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network. *Scientific reports*, 9(1): 1–11.

Zhu, J.; Chen, N.; and Peng, W. 2018. Estimation of bearing remaining useful life based on multiscale convolutional neural network. *IEEE Transactions on Industrial Electronics*, 66(4): 3208–3216.