Towards Autonomous PHM: An Application to Turboshaft Engine Torque Prediction

David He[1], Eric Bechhoefer[2], and Miao He[3]

[1]*University of Illinois-Chicago, Chicago, IL, 60607, USA*
davidhe@uic.edu

[2]*GPMS International Inc., Waterbury, VT, 05677, USA*
eric@gpms-vt.com

[3]*Siemens Corporation, Princeton, NJ 08540, USA*
miao.he@siemens.com

ABSTRACT

Most existing Prognostics and Health Management (PHM) systems are developed using supervised learning approaches, which depend heavily on labeled failure data. In practice, collecting extensive run-to-failure datasets from real-world assets is expensive, risky, and often impossible, particularly for safety-critical equipment such as turboshaft engines. This limitation makes it difficult for supervised PHM systems to maintain accurate models across different operating conditions and mission profiles, since they require frequent updates with newly labeled data.

To address these challenges and advance toward fully autonomous, on-board PHM, this paper proposes a self-supervised learning framework that continuously learns from abundant unlabeled operational data, adapts to new domains, and fuses heterogeneous sensor streams without relying on labeled failures. Self-supervised PHM systems extract features that are less domain-specific than those learned under supervision, enabling better generalization to unseen equipment and operating regimes. This capability is critical for moving from human-assisted PHM to fully autonomous decision-making in complex, variable, or inaccessible environments.

We introduce Time Series Bootstrap Your Own Latent (BYOL) with Task-Aware Augmentation (TS-BYOL-TAA), a novel self-supervised approach tailored for multivariate time-series data. TS-BYOL-TAA incorporates domain-informed augmentations to preserve task-relevant temporal structures while enhancing representation robustness. The method is evaluated in an autonomous PHM pipeline for turboshaft engine torque prediction, demonstrating improved adaptability and predictive accuracy compared to supervised baselines under domain shift and limited-label conditions. Results highlight the potential of self-supervised learning as a foundation for scalable, cross-platform autonomous PHM systems.

1. INTRODUCTION

Turboshaft engines are critical components in aerospace applications, particularly in rotorcraft, where high power density, reliability, and efficiency are essential for heavy-lifting and mission-critical operations. Modern helicopters, such as the Bell 407 equipped with the M250C47B engine, exemplify this advantage: delivering up to 804 horsepower while weighing only 273 lbs. Despite their robust design, these engines inevitably experience performance degradation over time due to factors such as compressor fouling, corrosion, erosion, and foreign object damage (FOD). Compressor fouling alone, caused by airborne contaminants adhering to compressor blades, can account for 70–85% of performance loss in some engines. Corrosion and erosion further compound this degradation, reducing airflow efficiency and mechanical integrity. Left undetected during operations, such performance loss can pose significant safety and mission-completion risks, especially in demanding environments.

Current Prognostics and Health Management (PHM) practices for turboshaft engines rely heavily on periodic, ground-based checks such as Power Assurance Checks, Health Indicator Tests (HIT), and Maximum Power Checks (MPC). While these procedures can detect performance deviations, they are inherently reactive, labor-intensive, and often fail to capture real-time health conditions during flight. In military and remote operations, these limitations are further amplified by environmental risks, limited access to maintenance personnel, and the operational cost of removing equipment from service. Automated, onboard health monitoring capable of detecting and predicting degradation in real time would not only improve safety and readiness but also enable condition-based maintenance scheduling and reduced lifecycle costs.

Most current PHM systems for aerospace engines are developed using supervised learning approaches, which require large quantities of labeled failure or degradation data. However, collecting real-world run-to-failure cases is expensive, risky, and sometimes impossible due to the rarity of catastrophic failures in modern, well-maintained fleets. Furthermore, supervised learning models are inherently domain-specific, requiring retraining when operating conditions, mission profiles, or engine variants change. This makes adaptation to new domains slow and inefficient, limiting the scalability of PHM systems across fleets and platforms.

To advance toward autonomous PHM, systems capable of self-monitoring, self-diagnosis, self-prognosis, and autonomous decision-making, there is a critical need for self-supervised learning (SSL) approaches. Unlike supervised learning, SSL can learn continuously from vast amounts of unlabeled operational data, extracting domain-agnostic representations that generalize well to unseen equipment and new mission scenarios. This capability enables PHM systems to adapt in real time to changing operational contexts and to fuse heterogeneous sensor streams for robust performance prediction. SSL-based PHM modules can thus serve as the foundation for moving from human-assisted analysis toward fully autonomous, on-board health management and decision-making.

In this paper, we propose Time Series Bootstrap Your Own Latent (BYOL) with Task-Aware Augmentation (TS-BYOL-TAA), a novel SSL framework specifically designed for multivariate time-series sensor data. The approach builds upon the BYOL paradigm but introduces task-aware augmentations that preserve mission-critical temporal dependencies while promoting invariance to irrelevant variations. The learned representations are both robust to domain shifts and sensitive to task-relevant changes, enabling accurate predictions even in low-label or cross-domain settings.

We evaluate TS-BYOL-TAA in the context of autonomous PHM for turboshaft engine torque prediction. Experimental results demonstrate that our method outperforms supervised baselines in predictive accuracy and adaptability, particularly under domain shift scenarios and with limited labeled data. These findings highlight the potential of self-supervised learning to overcome the data scarcity and adaptability challenges that currently limit the deployment of scalable, cross-platform autonomous PHM systems.

2. BACKGROUND AND RELATED WORK

2.1. Statistical Methods-Based Engine Power Prediction

In recent years, work on the automation of rotorcraft turboshaft engine performance monitoring and assessment using statistical approaches has been reported. Simon and Litt (2008) proposed a methodology to automate the Engine Torque Factor (ETF) calculation. Traditionally, the U.S. Army Black Hawk helicopters perform MPC to determine the ETF, requiring substantial pilot workload and posing safety risks in hostile environments. Their proposed method can be summarized as follows: (1) They primarily use curve-fitting techniques based on trend analysis. They partition the engine operating data into steady-state points using a steady-state data filter. This filter identifies segments of data from Health and Usage Monitoring Systems (HUMS) that reflect stable operating conditions, removing transient data that can introduce noise into performance assessment. (2) They calculate a residual (difference between actual and nominal engine performance) at steady-state points and update performance trends over time using exponential moving averages. This residual is used to track engine degradation. (3) To estimate the available power, they apply a least squares regression to fit the residuals and extrapolate engine performance at the limiting conditions (i.e., maximum turbine gas temperature or TGT). Their approach was applied to the T700-GE-701C engine in UH-60L Black Hawk helicopters, showing good agreement with manual power checks but highlighting areas for further validation and development. Bechhoefer (2024) presented a method for real-time monitoring of turboshaft engine performance, with a particular focus on providing engine margin and contingency power data to pilots in real-time. His method uses HUMS data to track engine parameters like measured gas temperature (MGT), torque (TQ), pressure altitude (PA), and outside air temperature (OAT). Utilizing these parameters, the system can alert pilots when the engine is underperforming, providing critical information for go/no-go decisions during missions. The statistical method presented by Bechhoefer (2024) can be summarized as follows: (1) Bechhoefer (2024) utilizes bicubic splines, a type of interpolation, to solve the inverse problem of estimating torque based on measured MGT, PA, and OAT. (2) The bicubic splines are used to create a model of engine performance at various operating conditions, interpolating between measured values to estimate torque. This spline-based approach allows for estimating power availability in real time during flight operations. (3) He also uses a linear regression model as a low-computation alternative to predict engine performance. This model is designed for real-time applications on embedded processors with limited computational resources.

While both methods in (Simon and Litt, 2008) and (Bechhoefer, 2024) offer significant benefits on computationally efficient and simple implementation for real-time applications, they may struggle with complex, non-linear relationships without significant preprocessing or transformation of data. These methods may potentially have a hard time addressing the following issues: (1) Engine-specific issue: The methods in both papers are highly reliant on engine-specific data. Each model is developed from scratch for a particular engine, making the systems slow to adapt to new engines without extensive data collection. (2) Limited generalization issue: The statistical approaches rely on local data and are less flexible in transferring insights from one engine to another. Each engine has its own performance model, and while these models work well for that engine, they cannot generalize well across different engines or operating conditions. In summary, while both papers utilize effective statistical techniques for their time, the integration of modern machine learning approaches could significantly enhance predictive accuracy, robustness, and real-time performance monitoring capabilities in rotorcraft turboshaft engines.

2.2. Transfer Learning-Based Engine Power Prediction

He, Bechhoefer, and Hess (2025) proposed a transfer learning-based method to automate the real-time prediction of turboshaft engine performance for rotorcraft. Traditional power checks are manual, often ground-based, and insufficient for capturing performance degradation during flight. The authors addressed these limitations using HUMS data and machine learning (ML) models, specifically transfer learning, to continuously estimate engine margin (a ratio of predicted vs. modeled torque) across different helicopter platforms and engine types. They used LSTM-based models trained on HUMS data and fine-tuned them for new engines and operating conditions, demonstrating strong generalization with minimal target-domain data. They investigated building torque prediction models using KNN, DNN, RNN, and LSTM architectures with HUMS inputs (e.g., MGT, PA, OAT, etc.) and showed that LSTM gave the best performance. They constructed the engine torque model using a deep LSTM trained on digitally extracted data from a power assurance check chart. The proposed method was applied to M250C47B engines from Bell 407 helicopters, showing that RMSE metrics demonstrated strong performance improvements using transfer learning, especially when dealing with new operating domains or unseen engine data.

2.3. LLMs for Engine Power Prediction

Tronconi, He, and Bechhoefer (2025) evaluated the potential of Large Language Models (LLMs) for predicting turboshaft engine torque, which is essential for helicopter safety and reliability. Their study investigated whether general-purpose and time-series-specific transformer models can serve as effective predictors using HUMS data. They used HUMS data from a Bell 407 turboshaft engine with 7 features (e.g., MGT, OAT, PA, IAS, Ng, Np, and torque), totaling 7954 samples to evaluate the following models: GPT-2 (fine-tuned), ChatGPT (prompt-based, zero-shot), TimeGPT (domain-specific time-series LLM), and standard trransformer-based time series model. Their evaluation results showed that while GPT-2 provided the best RMSE value, confirming LLMs' potential when fine-tuned on numerical data, TimeGPT gave strong RMSE accuracy and interpretability via variable importance. However, ChatGPT gave poor prediction RMSE accuracy but decent binary classification recall for low-torque conditions. The time series transformer provided a moderate performance.

2.4. Limitations of Related Work

In summary, despite their promises, the methods reviewed have some limitations when viewed through the lens of autonomous PHM. The model's performance depends heavily on the quality and completeness of HUMS data. HUMS data may be noisy, incomplete, or inconsistent across different platforms, which can limit scalability and reliability in real-time autonomy contexts. Though transfer learning reduces data requirements, it still requires fine-tuning with domain-specific samples. In a fully autonomous PHM setting (e.g., for newly deployed engines), there may be no time or opportunity to gather this tuning data. In general, most existing PHM systems are developed using supervised learning approaches, which depend heavily on labeled failure data. The challenges faced by these systems are limited labeled data for practical applications and inherent domain-specific models that are hard to generalize across different platforms and operating conditions. These challenges need to be addressed for developing an autonomous PHM system.

3. THE METHODOLOGY

The TS-BYOL-TAA proposed for turboshaft engine power prediction should be very similar to the MPC specified in the helicopter flight manual. For a typical MPC, the maximum allowable MGT will be manually determined based on the flight conditions of the helicopters and compared with the measured MGT. To automate this process, a numerical indicator is needed for continuous online monitoring of the engine power. In (Simon and Litt, 2008), ETF was used as the indicator of the actual power that an engine can produce relative to the rated power of the engine. Like that in (Simon and Litt, 2008), Bechhoefer (2024) used engine margin as the indicator, which is computed as the ratio of measured torque over the modeled torque at the maximum allowable MGT. In this paper, adapting the engine margin by Bechhoefer (2024) as the engine power indicator, TS-BYOL-TAA using HUMS data for turboshaft engine torque prediction is proposed. The flowchart of the proposed TS-BYOL-TAA using HUMS data for turboshaft engine torque prediction is shown in Figure 1.

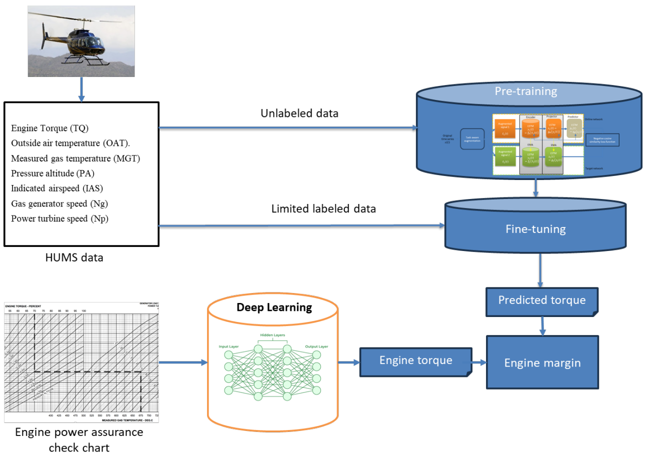

Figure 1. Flowchart of the proposed methodology.

As shown in Figure 1, unlabeled data collected by HUMS in the helicopters will be used to pre-train a TS-BYOL-TAA turboshaft torque prediction model. The purpose of the pre-training is to force the model to learn features that are invariant to data augmentations due to specific platforms or task operations. This results in robust representations that generalize better across input variations. Unlike other self-supervised contrastive learning frameworks such as Simple Framework for Contrastive Learning of Visual Representations (SimCLR) (Chen et al., 2020) or Momentum Contrast (MoCo) (He et al., 2020), BYOL doesn't need contrastive pairs or negative samples for pre-training. Therefore, it is flexible for BYOL to integrate other deep learning architectures like LSTM or transformers to improve its pre-training performance. After TS-BYOL-TAA pretraining, limited labeled data will be used to fine-tune the model for engine torque prediction. As the pretrained features generated from unlabeled data during pre-training act as a strong prior, fine-tuning with only a few labeled data often achieves better accuracy and faster convergence on downstream tasks like prediction. In the case of autonomous PHM applications, a pre-trained TS-BYOL-TAA engine torque prediction model can be used as a generalized model across multiple engine platforms. This generalized model can be easily adapted to a specific engine type with only a few labeled data points or even without any labeled data.

Engine torque will be predicted by the fine-tuned TS-BYOL-TAA turboshaft torque prediction model. The engine margin, as the engine performance indicator, will be computed as the ratio of the predicted torque to the modeled engine torque obtained from the engine power assurance check chart.

BYOL is a classic SSL method introduced by DeepMind in 2020 (Grill et al, 2020). Since then, it has been successfully applied to various self-learning problems (Feichtenhofer et al., 2021; Tian et al., 2021). It learns by matching two augmented views of the same sample using an online network and a target network. In BYOL, two augmented views of the same data are passed through an encoder and a projector. The goal is to make their embeddings similar. In SSL, labels are only used later, e.g., for fine-tuning in downstream tasks such as regression or classification.

Up to today, most of the BYOL applications have been focused on image or video representations. Limited application on multi-variant time series data is reported in the literature. In this paper, we introduce TS-BYOL-TAA, a novel BYOL-based framework specifically designed for multivariate time-series sensor data with task-aware augmentation. The structure of the TS-BYOL-TAA is presented in Figure 2.

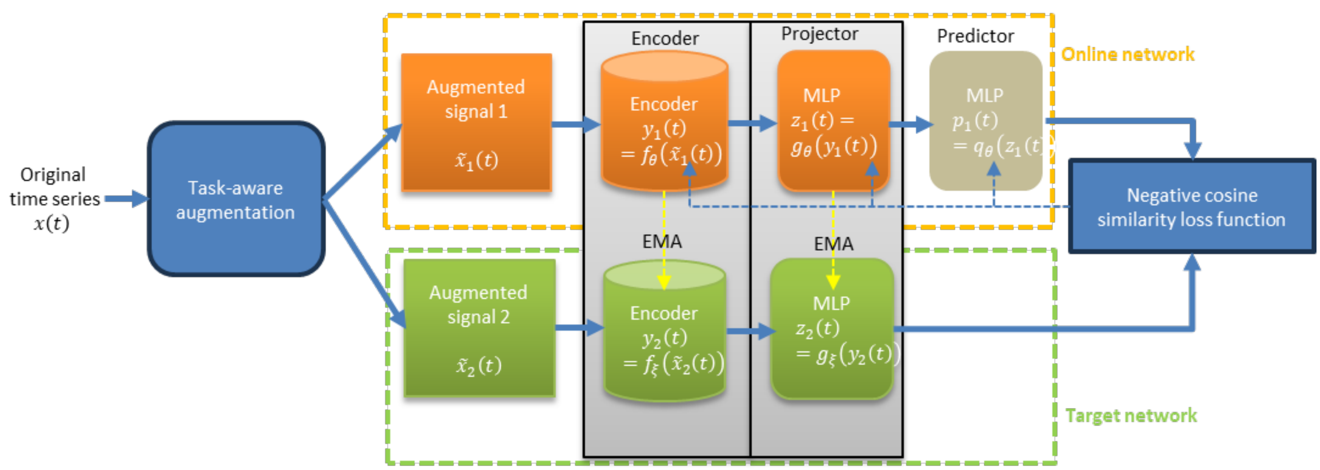

Figure 2. The structure of the TS-BYOL-TAA.

The time series task-aware augmentation of the raw input in the TS-BYOL-TAA model is implemented by integrating 3 commonly used data augmentation techniques for time series data: Gaussian noise, dropout-style masking, and time warping. Let x be the raw input data. Then x can be perturbed by adding noise ϵ as:

$$\tilde{x} = x + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2) \quad (1)$$

σ is the standard deviation of the noise and controls the noise scale. So, it is also called the noise scale. Perturbing the raw input data with Gaussian noise simulates input uncertainty and pushes the encoder to learn robust embeddings.

Dropout-style feature masking randomly set some feature values to zero to simulate missing or occluded data. It enforces feature-level invariance, i.e., the model should still align representations even when partial data is missing. It also helps prevent the model from relying on a small set of dominant features. Let $M \in \{0, 1\}$ be a random binary mask (Bernoulli). Then $\tilde{x}$, the augmented x, can be expressed as:

$$\tilde{x} = x \odot M, \text{ where } M \sim \text{Bernoulli}(p) \quad (2)$$

In Eq. (2), $p = 1 - dropout\ rate$, is defined as the keep probability and $\odot$ is the elementwise (Hadamard) product.

Time warping stretches or compresses the time axis in different regions of the signal, simulating temporal variation (like speed fluctuations in speech or machine behavior). Let the original time series be: $x(t) \in \mathbb{R}^F, t = 1, \ldots, T$. Time warping applies a smooth, nonlinear mapping $\phi(t)$ to the time indices:

$$\tilde{x}(t) = x\big(\phi(t)\big) \quad (3)$$

$\phi(t)$ is a monotonic warping function (ensures time doesn't go backward). The warped signal $\tilde{x}(t)$ has the same length T. But the samples come from warped time steps. Since $\phi(t)$ is usually non-integer, we use interpolation (e.g., linear or cubic) to compute $\tilde{x}(t)$:

$$\tilde{x}(t) = \text{Interproate}\left(x, \phi(t)\right) \quad (4)$$

This creates regions of temporal stretching (slowdown) and compression (speedup).

Now, we can integrate these 3 time series augmentation techniques as follows. Applying a time-warping function $\phi(t)$, we obtain $x\big(\phi(t)\big)$. Perturbing each feature in $x\big(\phi(t)\big)$ with Gaussian noise $\epsilon(t) \sim \mathcal{N}(0, \sigma^2)$, we generate $x\big(\phi(t)\big) + \epsilon(t)$. Finally, applying a binary mask $M(t) \sim \text{Bernoulli}(p)$ to simulate missing features, we obtain the augmented time series as:

$$\tilde{x}(t) = \Big(x\big(\phi(t)\big) + \epsilon(t)\Big) \odot M(t) \quad (5)$$

In general, the noise scale σ is fixed in TS-BYOL-TAA pretraining. However, this may limit the power of TS-BYOL-TAA. This is because in most applications of time series prediction, different features may have vastly different scales or variances. Adding fixed noise may overwhelm small-variance features and under-perturb large-variance ones. One strategy to overcome this limitation is the use of adaptive noise to scale perturbation proportionally to feature variability and maintain structure. In our integrated augmentation method, we set the noise scale as a learnable parameter of the TS-BYOL-TAA model to let the model learn per-feature noise scaling during training. In this way, our integrated time series data augmentation becomes the task-aware self-supervised augmentation.

As shown in Figure 2, the parameter set θ in encoder f_θ, projector g_θ, and predictor q_θ in the online network is updated through backpropagation using a negative cosine similarity loss function between the online network output vector p and another target network output z as:

$$\mathcal{L}_{loss} = \mathcal{L}(p_1, z_2) + \mathcal{L}(p_2, z_1)$$
$$\text{where } \mathcal{L}(p, z) = -\frac{p \cdot z}{\|p\|\|z\|} \quad (6)$$

The parameter set ξ in encoder f_ξ and projector g_ξ in the target network is updated through exponential moving average (EMA) as:

$$\xi \leftarrow \tau\xi + (1 - \tau)\theta \quad (7)$$

In Eq. (7), $\tau \in [0.99, 0.999]$ is a momentum coefficient.

4. THE APPLICATION CASE STUDY

4.1. The M250C47B Turboshaft Engine Data

In this paper, the real-world data from the Rolls-Royce M250C47 engine (Bechhoefer, 2024) is used to demonstrate the application of the developed methodology. The data was recorded by the HUMS in the Bell 407 helicopters. The HUMS data available to check the engine turbine health include OAT, MGT, PA, and TQ. According to the Bell 407 rotorcraft flight manual (Bell Helicopter-Textron, 2002), a power assurance check for the engine can be manually performed by a power assurance check chart (Figure 3) using these measurements to determine the maximum allowable MGT. The chart indicates the maximum allowable MGT for an engine meeting minimum Rolls-Royce specifications. The engine must develop the required torque without exceeding the chart MGT in order to meet the performance data contained in the chart.

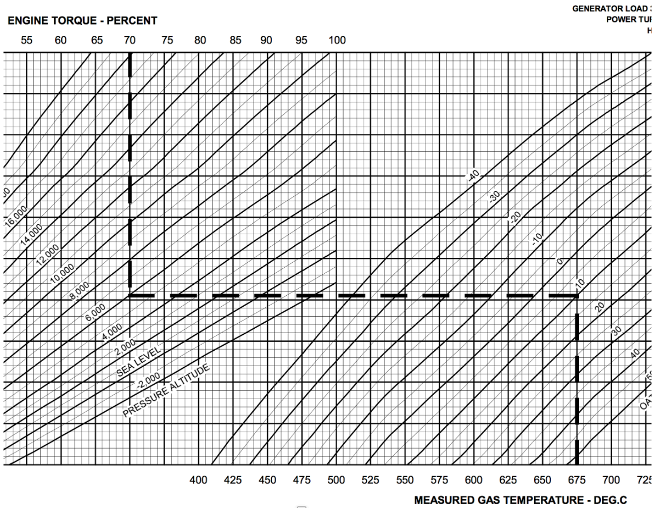

Figure 3. Power assurance check chart for M250C47 engine (Bell Helicopter-Textron, 2002).

The power assurance check chart in Figure 3 may be used to periodically monitor the engine performance.

To perform a power assurance check, all sources of bleed air, including engine anti-icing, will be turned off. A level flight at an airspeed of 85 to 105 KIAS or VNE, whichever is lower, will be established. A check may also be conducted in a hover prior to takeoff, depending on ambient conditions and gross weight (Bell Helicopter-Textron, 2002). For example, using the following recorded measurements: Hp = 6000 feet, OAT = 10°C, MGT = actual reading, TQ = 70%, one can enter the power assurance check chart at observed TORQUE (70%), proceed vertically down to intersect HP (6000 feet), follow horizontally to intersect indicated OAT (10°C), then drop vertically to read maximum allowable MGT of 675°C. If actual MGT is less than or equal to 675°C, engine performance equals or exceeds the minimum specification, and performance data contained in the flight manual can be achieved. The solid line in Figure 3 indicates the example power assurance check.

4.2. The Results

4.2.1. Performance of the Encoders

As mentioned earlier, BYOL-based self-learning is flexible in that it can take any deep learning architecture as its encoder since it doesn't need contrastive pairs or negative samples for pre-training. In our proposed TS-BYOL-TAA, while the projector and the predictor are a Multi-Layer Perceptron (MLP), the encoder can be either LSTM or GPT-2 because both LSTM and GPT-2 have been shown to be effective for time series prediction (He, Hess, and Bechhoefer, 2025; Tronconi, He, and Bechhoefer, 2025).

The effectiveness of LSTM and GPT-2 as the encoder in TS-BYOL-TAA was tested.

The LSTM encoder can be described as follows (He, Bechhoefer, and Hess, 2025). Define:

$\boldsymbol{X} \ni \mathbf{R}^{T\times d}$: the input sequence where T is the number of data points and d is the number of parameters at each data point

$h_t^{(l)}$: hidden state at data point t for layer $l = 1, \dots, L$

$c_t^{(l)}$: cell state at data point t for layer $l = 1, \dots, L$

$\boldsymbol{W}_f, \boldsymbol{W}_i, \boldsymbol{W}_o, \boldsymbol{W}_c$: weight matrices for the forget gate, input gate, output gate, and cell gate, respectively

b_f, b_i, b_o, b_c: biases for the respective gates

y: the target variable to be predicted

$\hat{y}$: the predicted value of the target variable

At each layer l, for each data point t, the LSTM operates with the following equations.

Compute the forget state value to decide which information to discard from the cell gate:

$$f_t^{(l)} = \sigma\left(W_f^{(l)}\left[h_{t-1}^{(l)}, x_t\right] + b_f^{(l)}\right) \tag{8}$$

Compute the input gate value to decide what information to store in the cell state:

$$i_t^{(l)} = \sigma\left(W_i^{(l)}\left[h_{t-1}^{(l)}, x_t\right] + b_i^{(l)}\right) \tag{9}$$

Compute the candidate values of the cell state:

$$\tilde{c}_t^{(l)} = \tanh\left(W_c^{(l)}\left[h_{t-1}^{(l)}, x_t\right] + b_c^{(l)}\right) \tag{10}$$

Update cell state by combining forget gate and input gate:

$$c_t^{(l)} = f_t^{(l)} \odot c_{t-1}^{(l)} + i_t^{(l)} \odot \tilde{c}_t^{(l)} \tag{11}$$

where $\odot$ denotes element-wise multiplication

After processing all the data points through the two LSTM layers, the last hidden state from the last layer $h_T^{(L)}$ is passed through a fully connected layer to make the final prediction as:

$$\hat{y} = W_{out} h_T^{(L)} + b_{out} \tag{12}$$

where W_{out} is the weight matrix of the output layer and b_{out} the bias.

To update the weight matrices of the LSTM network, the root mean square error (RMSE) is used as the loss function as:

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2} \tag{13}$$

In addition to the LSTM encoder, the Generative Pre-trained Transformer 2 (GPT-2) was also implemented as an encoder in the TS-BYOL-TAA model. GPT-2 with a transformer structure is a state-of-the-art pre-trained language model that has achieved impressive results in many Natural Language Processing (NLP) benchmarks (Radford et al., 2018; Radford et al., 2019; Brown et al., 2020). The structure of the GPT-2 encoder is provided in Figure 4.

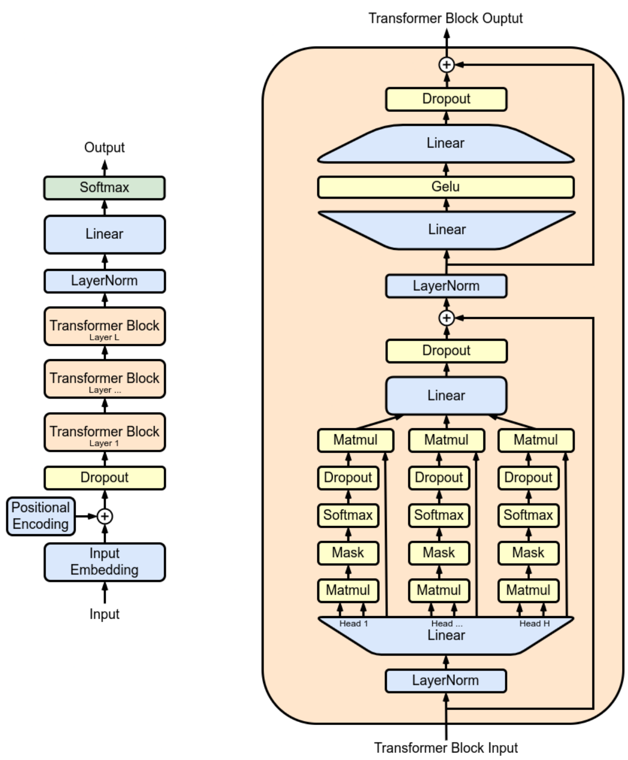

Figure 4. The structure of GPT2 (Wikipedia).

To pre-train the TS-BYOL-TAA model, HUMS data collected from two M250C47 engines in the Bell 407 helicopters were used. The dataset for each engine contains a total of 7954 data points, and each data point contains values for each of the 7 parameters: Engine Torque, OAT, MGT, PA, Indicated Airspeed (IAS), Gas Generator Speed (Ng), and Power Turbine Speed (Np). The ranges of the values for these parameters in the datasets are provided in Table 1.

Table 1. The HUMS dataset parameters and value range.

Parameter	Unit	Range
Engine Torque	%	[0.102, 93.445]
OAT	°C	[-2.086, 15.684]
MGT	°C	[454.633, 812.562]
PA	feet	[3348.425, 8699.112]
IAS	knot	[-0.186, 147.156]
Ng	rpm	[68.499, 99.889]
Np	rpm	[41.030, 102.988]

To pre-train and fine-tune the TS-BYOL-TAA models for predicting the engine torque using inputs OAT, MGT, PA, KIAS, Ng, and Np, 70% of the dataset was used for pre-training, 10% for fine-tuning the prediction head, and 20% for validation. The validation RMSE for each trained model is provided in Table 2.

Table 2. The RMSE of two different encoders.

Encoder	RMSE
LSTM encoder	0.0802
GPT-2 encoder	1.0666

4.2.2. The Feature Embeddings of TS-BYOL-TAA

TS-BYOL-TAA is basically a self-supervised representation learning method. During pretraining, it learns to produce useful embeddings by matching augmented views of the same input. To see how well the TS-BYOL-TAA model organizes the data in its latent space, we can visualize the feature embeddings learned by the TS-BYOL-TAA model after pretraining and compare them with the features of the raw input data. Figure 5 shows the comparison of the t-SNE embedding feature map after pretraining the TS-BYOL-TAA using LSTM encoder with that of the raw input data.

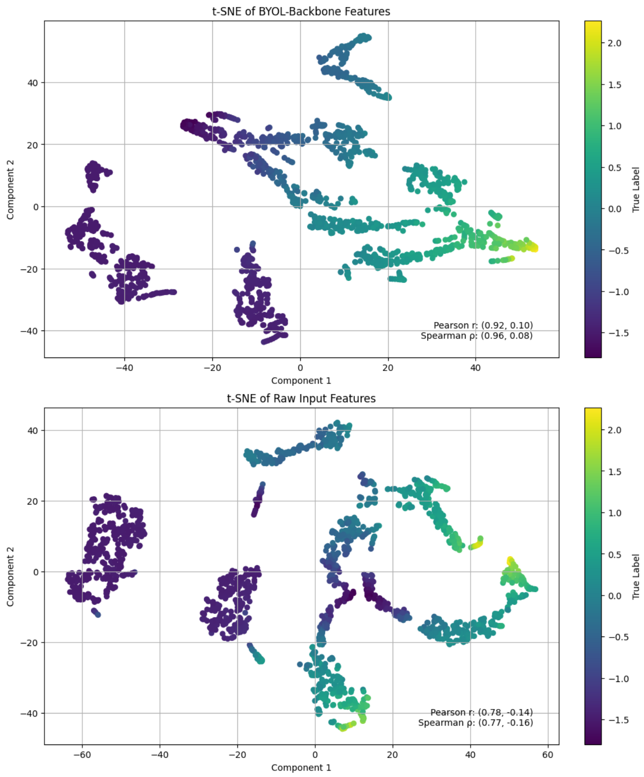

Figure 5. The t-SNE feature maps: top - embedding feature map after pretraining TS-BYOL-TAA with LSTM encoder; bottom - feature map of raw input data

From Figure 5, we can see that the latent features encode patterns related to the target, the engine torque. Here are the key observations: (1) Smooth color gradient. On the right-hand side (Component 1 ≈ 20–40), the points transition from green to yellow as the true label increases (≈50→90). On the left (Component 1 ≈ –50 to –30), the colors stay in the darker purples (≈10–20). That smooth change in hue tells us that nearby points in latent space tend to share similar target values. (2) Cluster formation. We can also see distinct "islands" of points, each island having a fairly tight color range. Those clusters suggest the backbone has learned to group inputs whose labels lie in the same range. (3) Implications for prediction. Because the embeddings already arrange themselves in a way that mirrors the target, our linear regression head should have a relatively simple mapping to learn, often leading to better generalization and faster convergence.

Note that quantitative summaries such as the Pearson correlation coefficient (r) and Spearman correlation coefficient (ρ) are also included for each t-SNE map in Figure 5. Basically, r and ρ give a numerical measure of how strongly the t-SNE components are related to the true target, engine torque. r takes a value between +1 and -1. $r = +1$ indicates a perfect positive linear correlation, $r = 0$ indicates no linear correlation, and $r = -1$ indicates a perfect negative linear correlation. ρ measures a monotonic relationship (not necessarily linear) and takes values between +1 and -1. ρ takes a similar interpretation as r, but based on the rank ordering of values. The quantitative summaries in Figure 4 indicate that the TS-BYOL-TAA backbone pretraining features correlate with the target, the engine torque more strongly than the raw input features. This confirms the effectiveness of the TS-BYOL-TAA method on the turboshaft engine torque data.

4.2.3. Domain Adaptation Performance

The major motivation to use TS-BYOL-TAA for turboshaft engine power performance prediction is to address the challenges faced by supervised learning-based PHM systems: limited labeled data and inherent domain-specific models. It is desired that the TS-BYOL-TAA model can predict performance degradation without requiring vast amounts of labeled data specific to each individual engine type. This allows for a more generalized model that can be applied across different helicopter platforms and engine configurations, significantly improving its practicality and scalability. To test the domain adaptation performance of the TS-BYOL-TAA model, we conducted two types of analysis: (1) We investigated how the TS-BYOL-TAA engine torque prediction model performs when the target domain shifts from one engine to another. (2) We investigated how the TS-BYOL-TAA engine torque prediction model performs when the target domain does not overlap with the source domain.

In the first analysis, we first tested how the TS-BYOL-TAA model pre-trained with engine 1 data to predict the torque of engine 2 performs and vice versa in comparison with supervised learning. In this test, only 10% of the labeled data was used for training in supervised learning and for fine-tuning for TS-BYOL-TAA model. The results of this test are provided in Table 3.

Table 3. Test results for the first analysis.

Models	RMSE	
	Eng1 -> Eng2	Eng2 -> Eng1
TS-BYOL-TAA with LSTM encoder	0.0728	0.0646
TS-BYOL-TAA with GPT-2 encoder	0.7812	0.7917
Supervised LSTM	0.0977	0.1121
Supervised GPT-2	1.2931	1.9358

As one can see from the results in Table 3, with the same small amount of labeled data, the TS-BYOL-TAA models gave better domain adaptation performance than the supervised learning-based counterparts.

In the second analysis, we tested how TS-BYOL-TAA models and the supervised learning models perform using the source domain data to predict the engine torque that is outside the source domain, i.e., the target domain. Figure 6 shows the plot of engine torque vs. MGT for engine 1 and the data portion of the source domain and target domain. As shown in Figure 6, the source domain data portion is inside the area specified by the red rectangular frame, i.e., all the data points that satisfy the following conditions will be used as the source domain data: Engine torque ≥ 20% and MGT ≥ 500 °C.

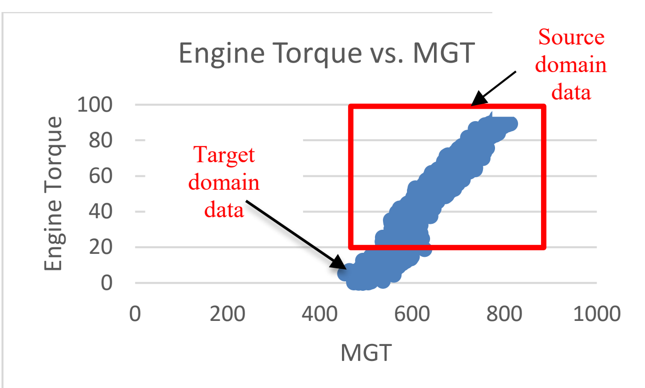

Figure 6. Engine torque vs. MGT plot for engine 1.

In this analysis, the TS-BYOL-TAA models were first pre-trained using the unlabeled source domain data and then were fine-tuned with 10% of the target domain data to predict the engine torque in the target domain. For the supervised learning models, 10% of the labeled data was used for training the models. The test results are provided in Table 4.

Table 4. Test results for the second analysis.

Models	RMSE
TS-BYOL-TAA with LSTM encoder	0.6533
TS-BYOL-TAA with GPT-2 encoder	1.5233
Supervised LSTM	1.7147
Supervised GPT-2	2.1326

As one can see from Table 4, again with the same small amount of labeled data, the TS-BYOL-TAA models gave better domain adaptation performance than the supervised learning-based counterparts.

5. CONCLUSIONS

This work presented TS-BYOL-TAA, a novel self-supervised learning framework for autonomous PHM in turboshaft engines, with a focus on torque prediction under domain shift and limited-label conditions. By integrating task-aware augmentations, i.e., Gaussian noise, dropout-style masking, and time warping, into a BYOL-based architecture, the method effectively preserved mission-critical temporal structures while improving robustness to irrelevant variations.

Applied to HUMS data from M250C47B turboshaft engines, TS-BYOL-TAA demonstrated clear advantages over supervised baselines. The learned representations exhibited stronger correlation with target variables, enabling more accurate torque prediction with significantly fewer labeled samples. Across both domain adaptation scenarios, cross-engine transfer and extrapolation beyond the source operating domain, the proposed framework consistently outperformed supervised LSTM and GPT-2 models, highlighting its capacity for generalized, cross-platform deployment.

The flexibility of TS-BYOL-TAA to incorporate different encoder architectures, such as LSTM or GPT-2, broadens its applicability to diverse time-series prediction tasks in aerospace PHM. Importantly, its ability to learn from abundant unlabeled operational data makes it a promising foundation for fully autonomous, on-board health monitoring systems that can adapt in real time to evolving operational environments without costly retraining.

Future work will focus on extending the approach to multi-modal sensor fusion, incorporating decision-making logic for autonomous maintenance actions, and validating performance across larger and more heterogeneous fleets. These advancements will further position self-supervised learning as a cornerstone technology for scalable, resilient, and intelligent PHM systems in safety-critical aerospace applications.

REFERENCES

Bechhoefer, E. (2024). A process for real-time performance monitoring of a turboshaft engine. *Proceedings of the 2024 PHM Society Annual Conference*, November 11–14, Nashville, TN, USA.

Bell Helicopter, Textron. (2002). *Bell 407 rotorcraft flight manual* (BHT-407-FM-1). Fort Worth, TX.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). Language models are few-shot learners. Advances in Neural Information Processing Systems, 33, 1877-1901.

Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. *Proceedings of the 37th International Conference on Machine Learning*, 119, 1597–1607. http://proceedings.mlr.press/v119/chen20j.html.

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2021). Self-supervised representation learning from videos: A BYOL approach. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15334–15344. https://doi.org/10.1109/CVPR46437.2021.01510.

Grill, J. B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., Kavukcuoglu, K., Munos, R., & Valko, M. (2020). Bootstrap your own latent: A new approach to self-supervised learning. *Advances in Neural Information Processing Systems, 33*, 21271–21284.

He, D., Bechhoefer, E., & Hess, A. (2025). Automated rotorcraft turboshaft engine performance prediction using a transfer learning approach. *IEEE Aerospace Conference*, Mar. 1–8, Big Sky, MT, USA.

He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738. https://doi.org/10.1109/CVPR42600.2020.00975.

Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. OpenAI Blog, 1(8), 9.

Simon, D. L., & Litt, J. S. (2008). *Automated power assessment for helicopter turboshaft engines* (NASA Technical Memorandum). Cleveland, OH: NASA Glenn Research Center.

Tian, Y., Chen, X., & Ganguli, S. (2021). Understanding self-supervised learning dynamics without contrastive pairs. *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 10268–10278.

Tronconi, A., He, D., & Bechhoefer, E. (2025). Evaluating large language models for turboshaft engine torque prediction. 2025 *PHM Society Annual Conference*, October 27–30, Bellevue, Washington, USA.