

Article

Pruning Quantized Unsupervised Meta-Learning DegradingNet Solution for Industrial Equipment and Semiconductor Process Anomaly Detection and Prediction

Yi-Cheng Yu ^{1,*} , Shiau-Ru Yang ¹, Shang-Wen Chuang ², Jen-Tzung Chien ¹  and Chen-Yi Lee ^{3,*}

¹ College of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Hsinchu City 300, Taiwan; zuzuyang0912.ee08@nycu.edu.tw (S.-R.Y.); jtchien@nycu.edu.tw (J.-T.C.)

² Advance Tech Co., Taipei City 114, Taiwan; swchuang@gmail.com

³ Department of Electronics Engineering, National Yang Ming Chiao Tung University, Hsinchu City 300, Taiwan

* Correspondence: yichengyu.ee06@nycu.edu.tw (Y.-C.Y.); cylee@nycu.edu.tw (C.-Y.L.)

Abstract: Machine- and deep-learning methods are used for industrial applications in prognostics and health management (PHM) for semiconductor processing and equipment anomaly detection to achieve proactive equipment maintenance and prevent process interruptions or equipment downtime. This study proposes a Pruning Quantized Unsupervised Meta-learning DegradingNet Solution (PQUM-DNS) for the fast training and retraining of new equipment or processes with limited data for anomaly detection and the prediction of various equipment and process conditions. This study utilizes real data from a factory chiller host motor, the Paderborn current and vibration open dataset, and the SECOM semiconductor open dataset to conduct experimental simulations, calculate the average value, and obtain the results. Compared to conventional deep autoencoders, PQUM-DNS reduces the average data volume required for rapid training and retraining by about 75% with similar AUC. The average RMSE of the predictive degradation degree is 0.037 for Holt–Winters, and the model size is reduced by about 60% through pruning and quantization which can be realized by edge devices, such as Raspberry Pi. This makes the proposed PQUM-DNS very suitable for intelligent equipment management and maintenance in industrial applications.

Keywords: anomaly detection; deep learning; prediction; pruning; edge computing; intelligent equipment management; meta-learning; retraining; semiconductor; unsupervised learning; vibration; quantization



Citation: Yu, Y.-C.; Yang, S.-R.; Chuang, S.-W.; Chien, J.-T.; Lee, C.-Y. Pruning Quantized Unsupervised Meta-Learning DegradingNet Solution for Industrial Equipment and Semiconductor Process Anomaly Detection and Prediction. *Appl. Sci.* **2024**, *14*, 1708. <https://doi.org/10.3390/app14051708>

Academic Editors: Janet Lin, Liangwei Zhang and Haidong Shao

Received: 20 January 2024

Revised: 14 February 2024

Accepted: 15 February 2024

Published: 20 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Research related to prognostics and health management (PHM) indicates that anomaly detection and prediction are important approaches to monitoring equipment faults and semiconductor process abnormalities. However, this type of detection often relies on subjective assessments performed by operators with prior experience. Automating the detection of equipment faults or semiconductor process anomalies is essential for reliable predictive maintenance and can potentially eliminate the need for manual monitoring. Moreover, interconnected intelligent monitoring systems play a crucial role in Industry 4.0, which focuses on artificial intelligence (AI)-driven factory automation.

In recent years, various deep-learning techniques have been introduced for anomaly detection and prediction in PHM, focusing on equipment faults (vibration and current anomalies) and semiconductor processes [1–10].

The related techniques include dense autoencoders (AEs) [11], convolutional AEs [12], and pretrained convolutional neural networks [13]. Although these deep-learning approaches exhibit excellent performance in anomaly detection, their widespread adoption in real factory settings remains limited. One major reason for the slow adoption is the high computational resource requirements of many deep-learning-driven anomaly detection

methods. These methods often lack practical considerations and system integration, ignoring factors such as multi-machine or multi-production line deployment, model retraining, optimization, and systematic integration. Consequently, their applicability in real factory environments is hindered.

Many studies have been conducted in the field of PHM. Pradeep et al. proposed that machine-learning techniques could be used to predict wafer defects with a random forest classifier, achieving an accuracy of over 93.62% [14]. This predictive maintenance approach enhanced the semiconductor manufacturing productivity. Nuhu et al. introduced synthetic data generation techniques that combined two missing value imputation methods and feature selection techniques [15]. This approach achieved an accuracy ranging from 99.5% to 100% when paired with the proposed machine-learning (ML) methods. Mao et al. introduced a novel deep AE (DAE) method that fused discriminative information with a gradient descent optimization approach [16]. This technique enhanced the numerical stability of the model in cases with limited training data. Abbasi et al. presented a series of highly compact deep convolutional AE network architectures that reduced the model size while maintaining a detection accuracy comparable to that of structures with over four million parameters [17]. Givnan et al. proposed an ML method for modeling and detecting anomalies during the operation of rotating machinery. This ML approach learned and generalized based on the fault severity to generate threshold values for anomaly detection [18].

A DAE model specifically designed for factory scenarios involving chillers was introduced that effectively distinguished between normal and abnormal vibration signals based on reconstruction differences [19]. Additionally, meta-learning was employed to improve the accuracy of the new sensor models with limited vibration data. For the new sensor model with fewer vibration data, the accuracy increased by about 33.50%. However, this method is mainly oriented to anomaly detection; it has not yet considered model retraining, anomaly prediction, lightweight models, edge computing, and integration with a complete intelligent management system.

Considering the aforementioned issues, this study proposes a Pruning Quantized Unsupervised Meta-learning DegradingNet System (PQUM-DNS) to address the real-world conditions of practical factories. This approach integrates five key features based on the actual needs of factories as illustrated in Figure 1.

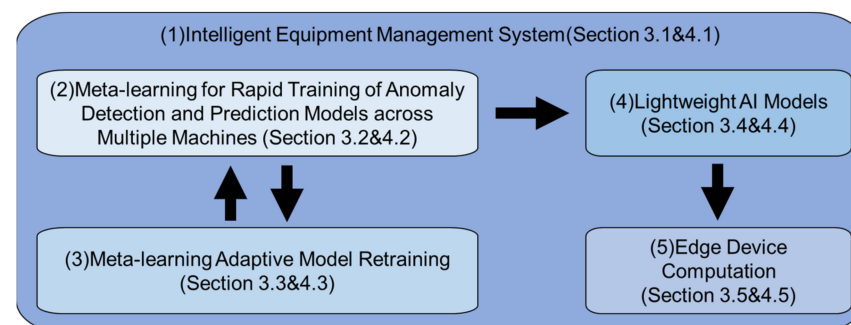


Figure 1. Overview of the proposed Intelligent Equipment Management System with related techniques highlighted in this paper.

(1) Intelligent Equipment Management System (Sections 3.1 and 4.1)

This system includes automated methods for vibration signal sensing, data transmission, data preprocessing, model training and retraining, anomaly detection, and prediction. Visual results are presented through dashboards and alert notifications are sent to onsite personnel and managers to facilitate timely problem solutions.

(2) Meta-learning for Rapid Training of Anomaly Detection and Prediction Models across Multiple Machines (Sections 3.2 and 4.2)

This approach rapidly establishes models for new machines or production lines with limited data by leveraging meta-learning and unsupervised learning through AEs, thereby achieving anomaly detection and prediction objectives.

(3) Meta-learning Adaptive Model Retraining (Sections 3.3 and 4.3)

Machine-specific models are adaptively retrained by employing meta-learning, quickly adjusting to the slow-changing characteristics of the machine over time and enabling long-term anomaly detection and prediction.

(4) Lightweight AI Models (Sections 3.4 and 4.4)

The proposed pruning and quantization compression model significantly reduces model size and conserves computational resources.

(5) Edge Device Computation (Sections 3.5 and 4.5)

Substituting traditional AI inference engines (IPC) with embedded Raspberry Pi systems enhances lightweight deployment, resource savings, cost reduction, and large-scale deployment feasibility.

2. Related Study

The following fundamental hardware, software techniques, and specifications are commonly used in semiconductor process and industrial scenarios for intelligent equipment management.

2.1. Vibration Signal Data Acquisition

(1) Device Sensors

One study collected real-time vibration data from equipment in an actual factory using an Advantech WISE-2410 LoRaWAN wireless sensor that integrated an ARMTM Cortex-M4 processor, LoRa transceiver, three-axis accelerometer, and temperature sensor [20]. It operated within a temperature range of -20°C to 85°C and was powered via USB.

(2) Vibration Data Feature Transformation

The raw vibration signals received from the sensors undergo feature transformation. The transformed features are critical for determining the vibration state and mainly include values such as velocity root mean square (RMS), acceleration RMS, acceleration peak, displacement kurtosis, displacement crest factor, displacement skewness, displacement peak to peak, and displacement deviation.

2.2. Vibration Signal Data of Chiller

Chillers are primarily utilized to build air conditioning systems. Chiller motors drive compressors and facilitate the exchange of heat and cold. In this study, vibration sensors were installed on a chiller motor to detect vibration values, as shown in Figure 2. Key vibration characteristics representing actual vibration measurements in a factory field were obtained through feature transformation [19].



Figure 2. Vibration sensor (in the red square) installed on the motor of a chiller machine.

Machine condition monitoring (MCM) was used to convert vibration signals into 44 key time- and frequency-domain physical and statistical feature values, as listed in Table 1 [21].

Table 1. Key time- and frequency-domain physical and statistical feature values of vibration signals transformed by MCM.

Feature Types	Feature
Time Domain (22 Features)	Clearance factor, max, mean, median, min, coefficient of variation, crest factor, local max, local min, max in range, frequency, impulse factor, kurtosis, percentile, peak to peak, RMS, shape factor, skewness, standard deviation, variance, X of max, X of min
Frequency Domain (22 Features)	Clearance factor, coefficient of variation, crest factor, frequency, impulse factor, kurtosis, local max, local min, max in range, max, mean, median, min, percentile, peak to peak, RMS, shape factor, skewness, standard deviation, variance, X of max, X of min

This study refers to ISO10816 [22] as shown in Figure 3 to determine the normal and abnormal vibration data of the chiller host motor. Results may vary depending on the size of the equipment where the sensor is installed.

VIBRATION SEVERITY PER ISO 10816					
Machine		Class I small machines	Class II medium machines	Class III large rigid foundation	Class IV large soft foundation
	in/s	mm/s			
Vibration Velocity Vrms	0.01	0.28			
	0.02	0.45			
	0.03	0.71		good	
	0.04	1.12			
	0.07	1.80			
	0.11	2.80		satisfactory	
	0.18	4.50			
	0.28	7.10		unsatisfactory	
	0.44	11.2			
	0.70	18.0			
	0.71	28.0		unacceptable	
	1.10	45.0			

Figure 3. Vibration Severity defined in ISO10816 [22].

2.3. Paderborn University Bearing Dataset

A condition-monitoring experimental bearing dataset based on vibration and motor current signals was used from Germany's Paderborn University, as shown in Figure 4 [23]. Experimental datasets were generated by installing different types of damaged ball bearings in the bearing test module. The setup consisted of healthy bearings with 1920 and 1600 entries for outer and inner ring damage, respectively. The most common bearing damage analysis [24] involves using motor current signals (MCSs) to convert time-domain signals into frequency-domain signals to observe the spectrum differences between normal and abnormal bearings as shown in Figure 5.

Feature engineering was applied to extract features from raw data, including common statistical, signal factor-related, fast Fourier transform (FFT), power spectral density (PSD)-related, and wavelet packet decomposition (WPD)-related features, as listed in Table 2.

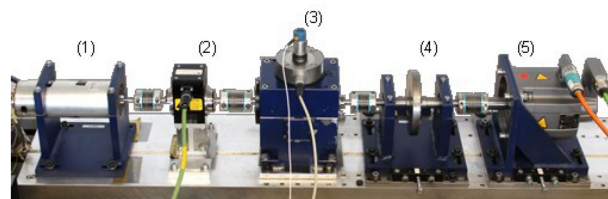


Figure 4. Germany's Paderborn University Bearing Dataset, consisting of: (1) motor, (2) torque measurement shaft, (3) rolling bearing test module, (4) flywheel, and (5) load motor.

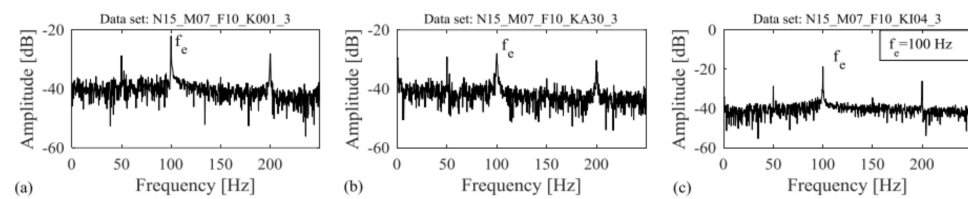


Figure 5. Frequency spectrum from MCS for (a) healthy bearing, (b) outer ring damage, and (c) inner ring damage [23].

Table 2. Feature values of vibration and current features after feature extraction.

Feature Types	Feature
Statistical feature	Mean, median, min, max, peak to peak, Std, variance, RMS, Absmean, Abslogmean, Meanabsdev, Medianabsdev, coefficient of variance, midrange
Signal factor-related feature	Shape factor, impulse factor, crest factor, clearance factor, skewness, kurtosis, entropy
FFT, PSD-related feature	1. Highest 3 FFT frequencies and values (6) 2. Highest 5 PSD frequencies (5) 3. Energy sum of FFT and PSD (2)
Wavelet packet decomposition (WPD)-related feature	Variance of 3 levels of WPD coefficient: Level 1 (2) Level 2 (4) Level 3 (8)

2.4. SECOM Semiconductor Analysis Dataset

The semiconductor manufacturing process utilizes semiconductor wafers as substrates and processes them through a series of steps. The main steps include: cleaning the wafer, depositing the film, cleaning after film formation, exposure, development, etching, inserting impurities, generating semiconductor properties, activation, assembly, and packaging. At each stage, sensors measure relevant parameters, including film thickness, size, resistance, temperature, etc. Through data analysis and machine learning, predictive maintenance, fault detection, process monitoring, and yield improvement are performed. These methods can also extract useful information from limited event records. Large amounts of data help solve predictive maintenance issues and build fault detection and diagnosis models. Through timely fault detection and diagnosis, downtime is reduced, costs are lowered, and product quality is enhanced [25].

Data were collected from a complex modern SECOM semiconductor manufacturing process that was under consistent surveillance through the monitoring of signals/variables collected from sensors or process measurement points. The dataset contained 1567 examples, 591 features, and 104 failures. Random forests were used to extract the 16 most important features for subsequent analysis to filter the key features because the SECOM dataset contained 591 features, as shown in Figure 6 [26–28]. For the SECOM dataset, the target column “−1” corresponds to a pass and “1” corresponds to a fail.

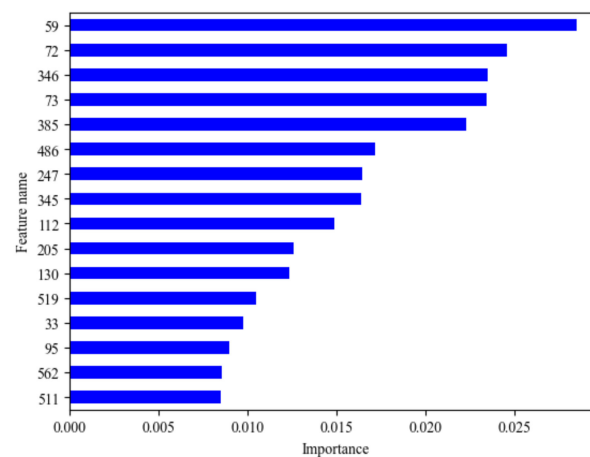


Figure 6. Extracting 16 key features from the SECOM dataset's 591 features using random forests.

2.5. Data Preprocessing

Data often encounter issues in practical scenarios, such as incomplete or missing data, noise, and outliers. These issues can disrupt the proper functioning of models. Hence, data preprocessing is essential for adjusting and manipulating data before applying analytical algorithms and preventing inaccurate judgments owing to flaws in the data. In this context, missing value deletion and statistical analysis were utilized to identify and eliminate values exceeding three standard deviations. This resulted in cleaner data, which facilitated subsequent analyses.

2.6. Equipment Process Degradation Level

The algorithm employed was an AE, which is a technique based on unsupervised learning. This approach involves computing the root mean square error (RMSE) by comparing the reconstructed output values of the model with the numerical input values. The calculated RMSE serves as an indicator of the equipment process degradation level, where smaller and larger values indicate healthier and poorer states, respectively [7,8].

2.7. Equipment Process Data Storage

A real-time PostgreSQL database was used to store the data collected from the sensors, including sensor names, equipment names, sensor registration times, operational data, and access to AI model training and inference. The inference results were stored and the database offered interfaces for other platforms to access the required data.

2.8. Area under Curve

Area under curve (AUC) is the area under the receiver operating characteristics (ROC) curve. AUC is a widely used evaluation metric in ML that assesses the performance of binary or multi-class classifiers [6,29]. Values range from 0 to 1, with a higher AUC value indicating better classifier performance. A notable advantage of AUC is its immunity to threshold variations, providing a comprehensive evaluation across different thresholds. This attribute makes AUC particularly robust in scenarios with imbalanced datasets. Figure 7 is an introduction to statistical principles related to AUC [30].

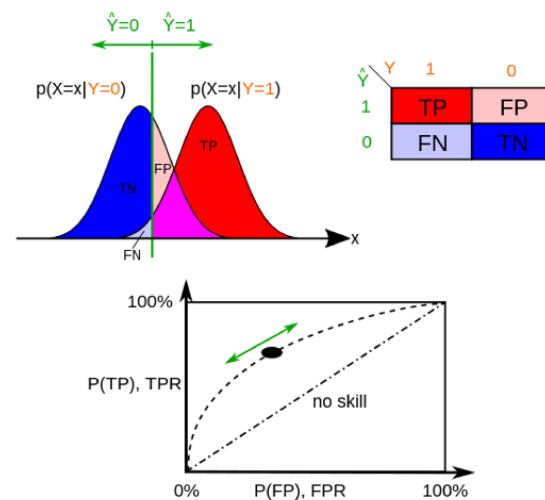


Figure 7. The results obtained from negative sample (left curve) overlap with the results obtained from positive samples (right curve). By moving (green arrow) the result cutoff value (vertical bar), the rate of false positives (FP) can be decreased, at the cost of raising the number of false negatives (FN), or vice versa (TP = True Positives, TPR = True Positive Rate, FPR = False Positive Rate, TN = True Negatives).

3. Proposed Method

3.1. Intelligent Equipment Management System

The proposed PQUM-DNS was integrated into an intelligent equipment management system for practical field applications. Signals were initially collected from the device sensors within the system before feature extraction transformation and data preprocessing, as shown in Figure 8. The feature extraction transformation involved converting raw vibration feature values into multiple key parameters related to machine health. Data preprocessing involved eliminating unnecessary empty and abnormal values to retain only the normal values required for unsupervised learning. The processed data were stored in a database (PostgreSQL) until there were sufficient accumulated data (e.g., 3000 records, adjustable). Subsequently, both AE and Holt–Winters algorithms were used to train the anomaly detection and prediction models, respectively. An inference was performed using anomaly detection and prediction models to generate results that indicated the degree of equipment or process degradation. The inference results were stored in a database and abnormal detection and prediction outcomes were sent to the visual dashboard of the intelligent management system, providing users with insights into the machine conditions. The system issued alert notifications to relevant personnel if the results of the abnormal detection and prediction exceeded the threshold of the AI equipment or process degradation. When retraining the model with a small amount of data using PQUM-DNS, the anomaly detection model can adapt after the first training of the Pretrain and Metatrain models, and then use a small amount of data to fine-tune retraining to obtain the new model.

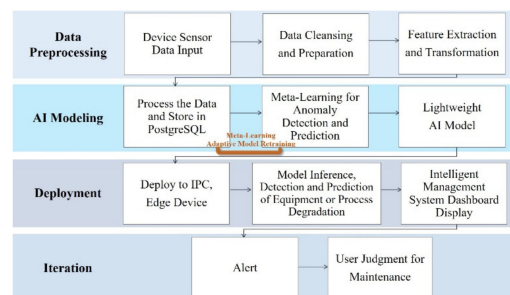


Figure 8. Flowchart of the proposed intelligent equipment management system.

3.2. Meta-Learning for Rapid Training of Multi-Machine Models for Anomaly Detection and Prediction

The rapid training of meta-learning multi-machine models for anomaly detection and prediction used in the PQUM-DNS was introduced in a previous study [19]. This method utilizes abundant data from numerous machines, trains the Metatraining model using the AE + meta-learning approach, and fine-tunes the Metatraining model with a small amount of data from new machines. This process yields a model adapted to a new machine, facilitating the inference for anomaly detection. In this study, the anomaly detection results were combined with prediction models to forecast future anomalies based on past anomaly detection outcomes. The following provides a brief introduction to the techniques employed.

(1) Meta-learning

Meta-learning is a technique aimed at enabling machine-learning systems to swiftly adapt to new tasks or environments [4]. Traditional machine-learning algorithms often require large amounts of labeled data to train models. In addition, it is necessary to collect and label substantial data for retraining when faced with new tasks. In contrast, the goal of meta-learning is to train a “learner” that is capable of rapidly learning new tasks from a small amount of labeled data. This approach typically relies on prior experience with numerous similar tasks and applies this experience to new tasks. These tasks can be expressed as:

$$\text{Learn } \theta \text{ such that } \Phi_i = f_{\theta}(D_i^{tr}) \text{ is good for } D_i^{ts} \quad (1)$$

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^n \log P(\phi_i | D_i^{ts}) \quad (2)$$

where

$$\Phi_i = f_{\theta}(D_i^{tr}) \quad (3)$$

$$D_{\text{meta-train}} = \{(D_1^{tr}, D_1^{ts}), (D_n^{tr}, D_n^{ts})\} \quad (4)$$

$$T_i \begin{cases} D_i^{tr} = \{(x_1^i, y_1^i), \dots, (x_k^i, y_k^i)\} \\ D_i^{ts} = \{(x_1^i, y_1^i), \dots, (x_l^i, y_l^i)\} \end{cases} \quad (5)$$

where T_i is a (meta-learning) task.

An illustration of the meta-learning method is shown in Figure 9 [31].

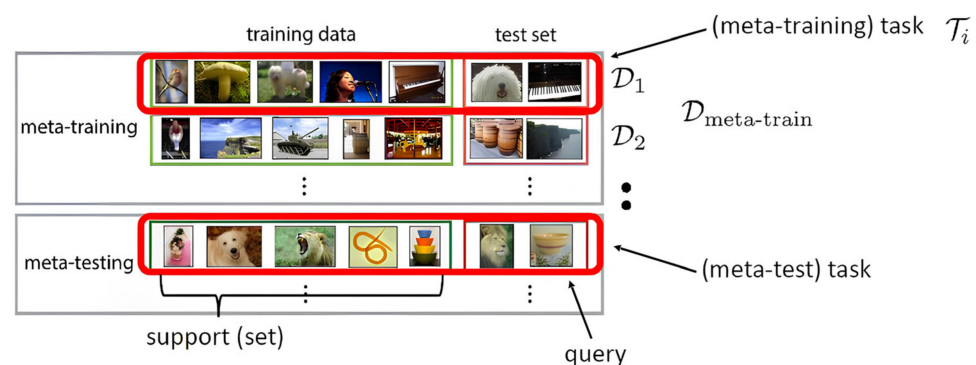


Figure 9. Illustration of the meta-learning method.

Various meta-learning models have been proposed for deep learning that are generally categorized as follows: learning good weight initializations, metamodels that generate the parameters of other models, and learning transferable optimizers. Model-agnostic meta-learning belongs in the first category and learns a good initial weight initialization to achieve fast adaptation to new tasks, enabling rapid convergence and fine-tuning of small-scale training samples [32].

(2) Anomaly Detection

The AI degradation level index was used to detect anomalies and the model was established using an unsupervised AE algorithm, as shown in Figure 10. This method involves constructing a model with normal data and applying an AE to compute the root mean square error (RMSE) between the input and output data, referred to as the reconstruction error. Here, the reconstruction error was defined as the AI degradation level index. A smaller value indicates a closer alignment between the model input and output values, leading to a better data reconstruction capability and a higher likelihood of normal equipment or processes.

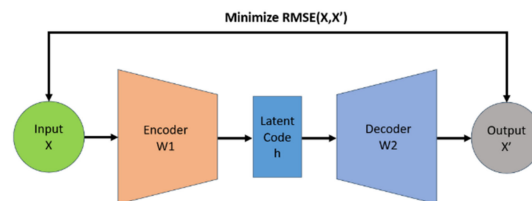


Figure 10. Structure of the autoencoder.

In practical applications, suitable threshold values are defined based on the conditions of the equipment or processes used for anomaly determination. These are expressed as:

$$h = g(W_1 X + b_1) \quad (6)$$

$$X' = f(W_2 h + b_2) \quad (7)$$

$$\text{Minimize } RMSE(X, X') \quad (8)$$

(3) Anomaly Prediction

Time-series algorithms can use historical data from the past to predict future trends. In this study, historical records of the health status of semiconductor manufacturing processes and machine equipment in the past can be used to predict future health status using time-series algorithms. The proposed anomaly detection models trained on the anomaly detection dataset could only detect current and past data anomalies. Therefore, an additional anomaly prediction model was required for future anomaly states. The historical data of the AI degradation level index obtained from anomaly detection were compared with various commonly used prediction models. Finally, the Holt–Winters algorithm was applied to the anomaly prediction model [7].

Various anomaly prediction algorithms are introduced below and the overview is in Table 3.

Table 3. Overview of time-series algorithms.

Algorithm	Overview (or Main Features)
Simple exponential smoothing (SES)	By placing a large weight on the most recent data, only two values and constants are needed to predict the next period of value
Holt (Holt's linear trend)	Holt's linear trend, which predicts trends in data, consists of a prediction equation and two smoothing equations for sequences that have a linear trend and no seasonality
Holt–Winters (Holt–Winters seasonal method)	The Holt–Winters method forecasts time series with both trends and seasonality
Autoregressive (AR)	The historical data of the variable itself are used to predict its own data, and the autoregression must meet the requirements of stationarity
Moving Average (MA)	A simple smoothing forecasting technique that calculates the sequence average of a certain number of items in turn according to the time-series data and the passage of time items to reflect the long-term trend
Autoregressive integrated moving average (ARIMA)	In the case analysis of non-stationary time series, the originally non-stationary time series becomes a stationary time series after many differences
Seasonal ARIMA (SARIMA)	ARIMA (differentially integrated moving average autoregressive) time-series-forecasting method with seasonal periodicity

(a) Simple Exponential Smoothing (SES)

This algorithm is used when there is no clear trend or seasonal pattern in the predictive data [33,34]. The prediction is calculated using weighted averages, meaning that the largest and smallest weights are associated with the most recently and least recently observed values, respectively. This is expressed as:

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1 - \alpha)y_{T-1} + \alpha(1 - \alpha)^2 y_{T-2} + \dots \quad (9)$$

where $\hat{y}_{T+1|T}$ denotes the one-step-ahead forecast for time $T + 1$, y_T denotes the most recent observation, and $0 \leq \alpha \leq 1$ denotes the smoothing parameter.

(b) Holt (Double Exponential Smoothing Method)

The Holt double exponential smoothing method is an extension of the simple exponential smoothing method that predicts trends in data [9]. This method is suitable for linear trending sequences without seasonal patterns and consists of one prediction and two smoothing equations that represent the level and trend components (l_t , b_t), which are respectively expressed as:

$$\hat{y}_{t+h|t} = l_t + hb_t \quad (10)$$

$$l_t = \alpha y_t + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (11)$$

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1} \quad (12)$$

where y_t and l_t denote the observed value and level at time t , respectively. b_t , h , α , and β^* denote the trend at time t , weight for the level ($0 \leq \alpha \leq 1$), and weight for the trend ($0 \leq \beta^* \leq 1$), respectively.

(c) Holt–Winters Forecasting (Triple Exponential Smoothing)

Holt–Winters forecasting, also known as triple exponential smoothing, is a method used to predict the behavior of time-series data that includes trends and seasonality. This algorithm considers three factors: the level l_t , trend b_t , and seasonal component s_t . It is effective for forecasting time-series data with seasonality patterns. There are two variations of this method: the additive and multiplicative models.

In the additive model, the components are expressed as:

$$\hat{y}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)} \quad (13)$$

$$l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (14)$$

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1} \quad (15)$$

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m} \quad (16)$$

In the multiplicative model, the components are expressed as:

$$\hat{y}_{t+h|t} = (l_t + hb_t)s_{t+h-m(k+1)} \quad (17)$$

$$l_t = \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (18)$$

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1} \quad (19)$$

$$s_t = \gamma \frac{y_t}{(l_{t-1} + b_{t-1})} + (1 - \gamma)s_{t-m} \quad (20)$$

where s_t , k , m , α , β^* , and γ denote the season at time t , integer part of $(h - 1)/m$, number of cycles/frequency of seasonality (e.g., four for quarterly data), level smoothing parameter, trend smoothing parameter, and seasonal smoothing parameters/weight for the season ($0 \leq \gamma \leq 1$), respectively.

In the additive model, the forecast value for each data element is the sum of the baseline, trend, and seasonality components. However, a multiplicative model is preferred when seasonal variations change proportionally to the level of the series.

(d) Autoregressive Model

The autoregressive (AR) model is a statistical method used to analyze time-series data that predict the future value of a variable using its own historical data [35]. AR is an evolution of linear regression analysis, where it analyzes the relationship between parameter x and its own past value instead of analyzing the relationship between the parameter x and dependent variable y . This is expressed as:

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t \quad (21)$$

where y_t , C , p , ϕ_i , and ε_t denote the stationary time series, constant term, autoregressive order, non-zero autocorrelation coefficients, and independent error term, respectively.

(e) Moving Average

Moving average (MA) is a simple smoothing prediction technique used for time-series data that calculates a moving average over a certain number of terms to reflect long-term trends [36]. However, it is difficult to discern the development trend when time-series data are influenced by periodic and random variations causing large fluctuations. Using MAs can eliminate these influences and reveal the direction and trend of the events, which is expressed as:

$$y_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \quad (22)$$

where y_t , μ , q , and θ_i denote the stationary time series, mean of the sequence, moving average order, and non-zero autocorrelation coefficients, respectively.

(f) Autoregressive Integrated Moving Average

Autoregressive integrated moving average (ARIMA) is an evolution of the AR, MA, and autoregressive moving average models. This approach is used to analyze non-stationary time-series data by transforming them into stationary data through differencing [37]. This method is employed when dealing with non-stationary time-series data that exhibit a changing mean and variance over time. A new stationary time series can be obtained by using the differences in the data, and a suitable probabilistic model can be derived from historical data to represent the dependence between time and data. ARIMA can be expressed as ARIMA (p, d, q), where p , d , and q denote the autoregressive, differencing, and moving average orders, respectively. Furthermore:

$$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (23)$$

where ϕ_i and θ_i denote non-zero autocorrelation coefficients.

(g) Seasonal Autoregressive Integrated Moving Average

The seasonal autoregressive integrated moving average (SARIMA) model incorporates seasonal factors into the ARIMA model [38]. Generally, the SARIMA model is denoted as SARIMA (p, q, d)(P, Q, D) s , where s , P , Q , D denote the seasonal period and seasonal autoregressive, seasonal moving average, and seasonal differencing orders, respectively. This is expressed as:

$$\left(1 - \sum_{i=1}^p \phi_i B^i\right) \left(1 - \sum_{i=1}^P \Phi_i B^{im}\right) (1 - B)^d (1 - B^{Ds}) y_t = \left(1 + \sum_{i=1}^q \theta_i B^i\right) \left(1 + \sum_{i=1}^Q \Theta_i B^{im}\right) \varepsilon_t \quad (24)$$

where B denotes the lag operator, and Φ_i and Θ_i denote non-zero constants.

3.3. Meta-Learning Adaptive Model Retraining

Meta-learning was employed to develop an adaptive method for retraining machine-specific models [4]. This approach enables models to quickly adjust to the slow changes observed in each machine over time, thereby achieving prolonged anomaly detection. The concept of rapid training models for different machine devices was extended to different time segments by combining the background technique of AEs with meta-learning.

The data were segmented into three intervals based on chronological order: Pretrain (older and long-running machine data), Metatrain (newer and long-running machine data), and Fine-tune (latest operational data). Leveraging the principles of meta-learning, Pretrain and Metatrain data were used to train a generalized anomaly detection model, whereas Fine-tune adapted the model to the most recent machine conditions. This approach automatically trains models that adapt to data changes over time.

Model retraining was based on the operational conditions of the factory to maintain the effectiveness of the anomaly detection model. The historical data were also divided into Pretrain, Metatrain, and Fine-tune segments by utilizing seven days of equipment operation data, as shown in Figure 11. This process led to the training of an anomaly detection model that could assess the degree of equipment degradation, as shown in Figure 12. Additionally, predictive algorithms were applied to forecast equipment degradation over the next seven days. The ultimate goal was to achieve continuous automatic updates for anomaly detection and prediction.

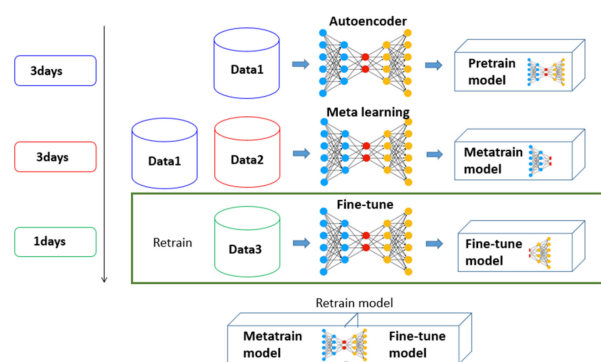


Figure 11. Historical data of seven days segmented into Pretrain, Metatrain, and Fine-tune for training an anomaly detection model to detect equipment degradation.

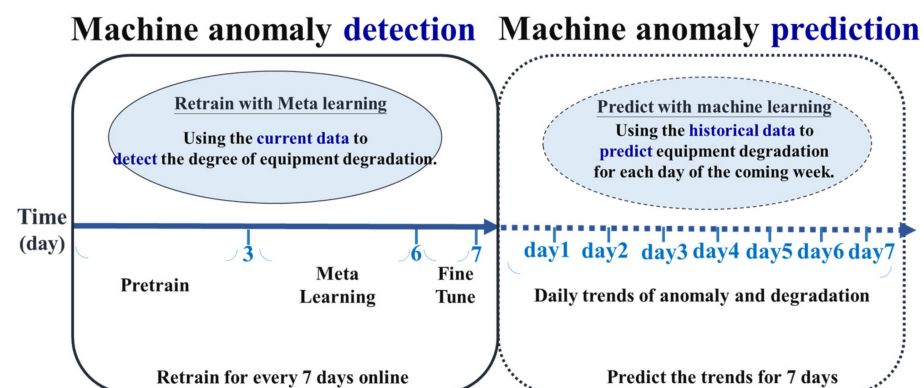


Figure 12. Automatic model retraining based on time intervals, with the retrained model used for predicting anomalies in the next seven days.

3.4. Lightweight AI Model

A pruning and quantization-based meta-learning anomaly detection model was introduced owing to the possibility of certain neural neuron weights being small or negligible

during the retraining process of the neural network of the proposed model. This approach significantly reduced the model size and enhanced the computational speed.

(1) Model Pruning

Deep-learning neural network models often contain redundant parameters, with many neuron weights approaching zero. Model pruning involves removing these neurons while preserving the same model expressive capability. Model pruning retains the essential weights and parameters, reducing the number of connections between the neural network layers, as shown in Figure 13 [39]. This reduction helps to decrease the number of parameters involved in the calculations, thereby lowering the computation requirements. By maintaining the performance of the model, this approach reduces the storage space, lowers computational costs, and accelerates the training process.

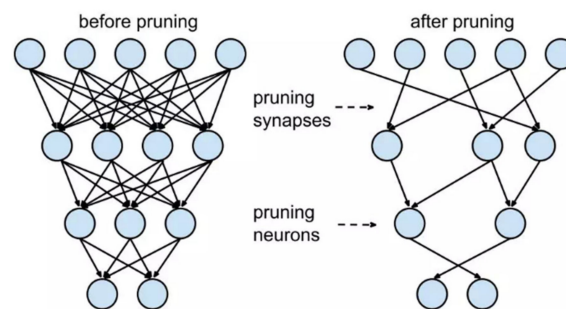


Figure 13. Model pruning retains important weights and parameters while reducing the number of connections between neural network layers.

Pruning algorithms typically employ a three-stage pipeline: training, pruning, and fine-tuning. The weight adjustment process in the three-step training pipeline for pruning is shown in Figure 14. The model's weights are trained, pruning techniques are applied to remove neurons with weights approaching zero, and the model is fine-tuned to adjust the remaining weights to approximate the performance of the original model. This iterative process helps align the performance of the model with that of the original model.

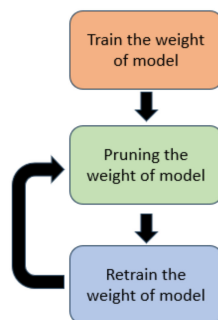


Figure 14. Weight adjustment process in the three-step training pipeline for pruning.

(2) Model Quantization

The principle of quantization involves reducing the precision of the bits used to represent model parameters (typically 32-bit floating-point (float32) numbers) [40]. This approach results in smaller model sizes and faster computations. Model quantization involves approximating the continuous values (or a large number of possible discrete values) of the floating-point model weights with a limited set of discrete values (usually 8-bit integer (int8) numbers) at a lower inference accuracy loss, as shown in Figure 15 [40]. A lower-bit data type is used to approximate the finite-range floating-point data, which

leads to a reduced model size, decreased memory consumption, and faster inference speed. The calculations are expressed as:

$$Q = \frac{R}{s} + z \quad (25)$$

$$s = (R_{max} - R_{min}) / (S_{max} - S_{min}) \quad (26)$$

where R , Q , z , and s denote the real floating-point value, fixed-point value after quantization, fixed-point value after quantization of the 0 floating-point value, and minimum scale that can be represented after fixed-point quantization, respectively.

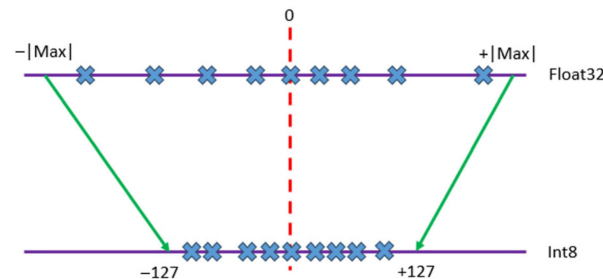


Figure 15. Model quantization: conversion from float32 to int8.

The model-pruning approach is applied using weight sparsity, where weights close to zero are removed from the original model. Subsequently, the model is retrained to adjust its performance. Additionally, quantization techniques are combined to convert the weights from float32 to int8, thereby significantly reducing the model size and enhancing the computational speed.

3.5. Edge Device Computing

The pruned and quantized models are deployed on embedded systems, such as a Raspberry Pi, replacing traditional IPCs. This lightweight approach conserves resources, reduces costs, and facilitates a large-scale deployment. Edge devices can analyze data and promptly alert onsite maintenance personnel in industrial scenarios where certain equipment is located in hard-to-reach locations [10,18]. This enables real-time responsiveness to the equipment conditions, allowing immediate intervention and preventing downtime.

Benefits of edge computing are as follows:

- Provides rapid real-time reflection of situations, enabling onsite personnel to detect anomalies promptly and take immediate action.
- Solves bandwidth issues in cloud and edge transmissions because edge devices only need to send inference results back to the control centers.
- Addresses cybersecurity concerns, protecting against network attacks that could lead to factory shutdowns.
- Reduces energy consumption because lightweight edge-computing models conserve power.

4. Results and Discussion

4.1. Intelligent Equipment Management System

Automated methods are used for vibration signal sensing, data transmission, data preprocessing, model training, and retraining. The results are visually presented through dashboards. The system utilizes AI degradation level values to detect anomalies and sends warning notifications for timely handling by managers and onsite personnel. A comparison between this system and traditional methods is presented in Table 4.

Table 4. Comparison of intelligent equipment management system and traditional approaches.

Issue	Intelligent Management System	Traditional Management System
Intelligent management	Integrate and automate equipment alarms, display visualization results in a Kanban style, and send warning notices to facilitate timely handling of problems by managers on site.	A manager will only be notified of a situation by onsite personnel when an abnormality occurs in the equipment or the production line stops, thus not dealing with the situation in a more timely manner.
Meta-learning anomaly detection and prediction	Apply meta-learning to quickly train AI models for automatic detection and prediction of new equipment anomalies for preventive equipment maintenance.	New machine models require a great deal of data to train, personnel need to confirm the condition of the equipment from time to time, and preventative maintenance cannot be performed in advance.
Meta-learning adaptive modeling with retraining	Meta-learning can be used to quickly adapt to the characteristics of machines that change slowly over time, thus realizing the purpose of model updating over a long period of time.	The model is retrained by AI analysts when an abnormality occurs in the model, which is labor-intensive and increases risk to the equipment.
Lightweight quantitative AI models	Dramatically reduces the size of the model and increases the speed of the operation.	Larger models consume more hardware space for storage and run more slowly.
Edge computing	It can be lightweight, save resources, reduce cost, and achieve the purpose of large-scale parts.	Larger PC computing devices are bulky, heavy, costly, energy-intensive, lack mobility, and are difficult to deploy on a large scale.

4.2. Meta-Learning for Rapid Training of Multi-Machine Models for Anomaly Detection and Prediction

For the sake of versatility, meta-learning is applied to multi-machine anomaly detection and prediction models with different datasets, such as factory chiller vibration data, publicly available datasets for analyzing the vibration and current of Paderborn bearings, and SECOM semiconductor analysis.

(1) Rapid Training of Multi-Machine Models for Anomaly Detection

The proposed PQUM-DNS method is compared with the general DAE method. PQUM-DNS achieves similar AUC values with minimal data compared to those of the DAE method when used to train models for new machines. This is because PQUM-DNS used meta-learning to train a versatile model applicable to various conditions (Metatraining model). Therefore, training a new machine model only requires a small amount of data for fine-tuning, resulting in a rapidly adaptable anomaly detection model. Test results for different data types, such as chiller vibration, SECOM, and Paderborn current and vibration datasets are presented in Table 5 and Figure 16. Compared with the DAE, PQUM-DNS reduces the required data for training new machine models on average by approximately 75%, with a decrease in AUC performance of only 0.35%, which is a very close AUC value.

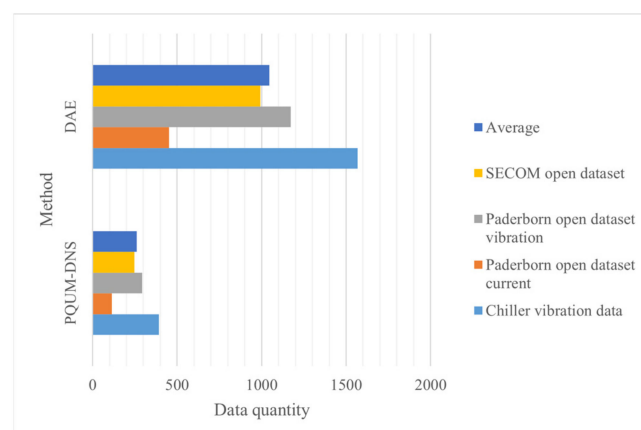
**Figure 16.** Comparison of PQUM-DNS and DAE training data.

Table 5. Comparison of PQUM-DNS and DAE training performance for different data types.

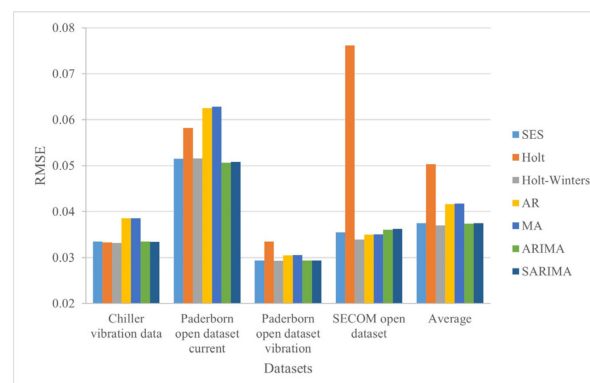
Training					
	Chiller Vibration Data	Paderborn Open Dataset Current	Paderborn Open Dataset Vibration	SECOM Open Dataset	Average
PQUM-DNS					
AUC (%)	99.99	92.73	99.81	68.97	90.38
Data quantity	392	113	293	248	262
DAE					
AUC (%)	99.99	97.03	99.73	66.14	90.73
Data quantity	1569	452	1172	993	1047
Improvement					
AUC Increase (%)	0.00	−4.3	0.07	2.83	−0.35
Data quantity Decrease (%)	75.02	75	75	75.03	75.01

(2) Anomaly Prediction:

PQUM-DNS detects machine degradation levels through anomaly detection and evaluated these levels using various prediction algorithms. Seven prediction algorithms are compared and one based on RMSE calculations is selected based on the predicted and actual values. The performances of these algorithms on different datasets are presented in Table 6 and Figure 17. The Holt–Winters algorithm demonstrates the best performance, with the lowest RMSE value of approximately 0.037, making it the chosen algorithm for anomaly prediction in the PQUM-DNS.

Table 6. Comparison of PQUM-DNS prediction algorithm performance on different datasets.

RMSE					
Algorithm	Chiller Vibration Data	Paderborn Open Dataset Current	Paderborn Open Dataset Vibration	SECOM Open Dataset	Average
SES	0.03350	0.05153	0.02934	0.03547	0.03746
Holt	0.03330	0.05824	0.03348	0.07614	0.05029
Holt–Winters	0.03315	0.05161	0.02928	0.03395	0.03700
AR	0.03855	0.06250	0.03052	0.03500	0.04164
MA	0.03853	0.06289	0.03057	0.03505	0.04176
ARIMA	0.03349	0.05062	0.02933	0.03606	0.03737
SARIMA	0.03347	0.05088	0.02932	0.03626	0.03748

**Figure 17.** Comparison of PQUM-DNS prediction algorithm performance on different datasets.

4.3. Meta-Learning Adaptive Model Retraining

An adaptive method is employed for retraining the machine model using meta-learning. This enabled the model to adapt quickly to gradual changes over time, thereby facilitating long-term anomaly detection. PQUM-DNS chronologically segments data from the same machine and fine-tunes the model using the latest data, thereby achieving a model suited to the machine's latest condition. Unlike the general DAE method that requires retraining with all data, PQUM-DNS significantly reduces the amount of data needed for retraining, as shown in Table 7. This is because PQUM-DNS already trains a versatile meta-learning model using past data (a meta-trained model), enabling efficient fine-tuning with a small amount of new machine data.

Table 7. Comparison of PQUM-DNS and DAE retraining performance for different data types.

Retraining					
	Chiller Vibration Data	Paderborn Open Dataset Current	Paderborn Open Dataset Vibration	SECOM Open Dataset	Average
PQUM-DNS					
Retraining data quantity (fine-tune)	586	54	50	248	235
AUC	99.99	94.68	99.99	62.6	89.32
DAE					
Retraining data quantity	2346	219	204	993	941
AUC	99.99	96.17	99.99	66	90.54
Improvement					
Retraining data quantity decrease	75.02	75.34	75.49	75.03	75.07
AUC increase	0.00	−1.49	0.00	5.15	1.35

4.4. Lightweight AI Model

PQUM-DNS drastically reduces the model size while maintaining a similar AUC performance to that of the non-lightweight DAE model. This is achieved by removing the near-zero weights from the original DAE-trained model and compressing the model data format from float32 to int8. Consequently, the model size is significantly reduced. The application of PQUM-DNS to various datasets demonstrates that the lightweight model size is reduced by approximately 60% with AUC performance maintained at similar levels, as shown in Table 8 and Figure 18.

Table 8. Comparison of PQUM-DNS and DAE model sizes for different data types.

Model Size					
	Chiller Vibration Data	Paderborn Open Dataset Current	Paderborn Open Dataset Vibration	SECOM Open Dataset	Average
PQUM-DNS					
Model size (bytes)	5776	6969	5973	5866	6146
AUC	99.31	93.16	99.34	69.69	90.38
DAE					
Model size (bytes)	18767	14765	19466	11866	16216
AUC	99.31	93.16	99.34	69.69	90.38
Improvement					
Model size decrease	69	53	69	51	60
AUC increase	0.00	0.00	0.00	0.00	0.00

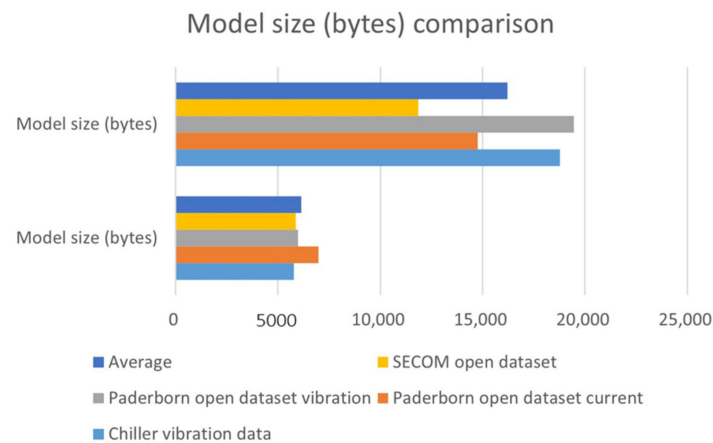


Figure 18. Comparison of PQUM-DNS and DAE model sizes for different data types.

4.5. Edge Device Computing

The PQUM-DNS, with its reduced and compressed model size, is well suited for lightweight edge-computing devices. Therefore, it is applied to replace traditional IPCs with embedded systems, such as a Raspberry Pi. This substitution reduces the size and weight, conserves resources, lowers costs, and supports large-scale deployments.

5. Conclusions

This study proposes a new PQUM-DNS model, which is an intelligent device management system that combines pruning, quantization, meta-learning, anomaly detection, prediction using AEs, adaptive model retraining, and edge inference. This system effectively reduces the manual labor, provides fault notifications, prevents downtime, decreases model computational resources, accelerates the model inference speed, and enables edge inference.

The system is suitable for various factory scenarios and types of machine equipment and process states. Compared with general DAEs, the system achieves a similar AUC while reducing the training data by approximately 75%. The average RMSE of the predictive degradation degree is 0.037 for Holt–Winters, retraining is conducted using 75% fewer data with similar AUC performance, and the model size is reduced by approximately 60% through pruning and quantization. The proposed system can be deployed on lightweight edge devices, such as a Raspberry Pi, enabling real-time anomaly detection and prediction. The system demonstrates superior performance, thereby realizing intelligent equipment management and maintenance.

Author Contributions: Conceptualization, Y.-C.Y.; methodology, Y.-C.Y. and S.-W.C.; software, Y.-C.Y.; validation, Y.-C.Y.; formal analysis, Y.-C.Y.; investigation, Y.-C.Y.; resources, C.-Y.L. and S.-W.C.; data curation, Y.-C.Y.; writing—original draft preparation, Y.-C.Y.; writing—review and editing, C.-Y.L. and S.-R.Y.; visualization, Y.-C.Y.; supervision, C.-Y.L., J.-T.C. and S.-W.C.; project administration, C.-Y.L., Y.-C.Y. and S.-R.Y.; funding acquisition, Y.-C.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: <https://mb.uni-paderborn.de/kat/forschung/kat-datacenter/bearing-datacenter/data-sets-and-download>, accessed on 1 June 2023 (Paderborn University Bearing Dataset) and <https://archive.ics.uci.edu/dataset/179/secom>, accessed on 1 June 2023 (SECOM Dataset).

Acknowledgments: Thanks for the data support from Advantech Taipei headquarters and the technical support from SI2 Laboratory at National Yang Ming Chiao Tung University.

Conflicts of Interest: The authors declare no conflict of interest.

Nomenclature

AE	Autoencoder
AI	Artificial Intelligence
AR	Autoregressive
ARIMA	Autoregressive Integrated Moving Average
AUC	Area under Curve
DAE	Deep AE
FFT	Fast Fourier Transform
IPC	Industrial Personal Computer
LoRaWAN	Long-Range Wide-Area Network
MA	Moving Average
MCM	Machine Condition Monitoring
MCS	Motor Current Signal
ML	Machine Learning
PHM	Prognostics and Health Management
PQUM-DNS	Pruning Quantized Unsupervised Meta-learning DegradingNet Solution
PSD	Power Spectral Density
RMS	Root Mean Square
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristics
SARIMA	Seasonal Autoregressive Integrated Moving Average
USB	Universal Serial Bus
WPD	Wavelet Packet Decomposition

References

1. Tung, F.; Mori, G. Clip-q: Deep network compression learning by in-parallel pruning-quantization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7873–7882.
2. Frantar, E.; Alistarh, D. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 4475–4488.
3. Hu, P.; Peng, X.; Zhu, H.; Aly, M.M.S.; Lin, J. Opq: Compressing deep neural networks with one-shot pruning-quantization. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 7780–7788. [\[CrossRef\]](#)
4. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-learning in neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *44*, 5149–5169. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Givnan, S.; Chalmers, C.; Fergus, P.; Ortega-Martorell, S.; Whalley, T. Anomaly detection using autoencoder reconstruction upon industrial motors. *Sensors* **2022**, *22*, 3166. [\[CrossRef\]](#)
6. Bradley, A.P. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* **1997**, *30*, 1145–1159. [\[CrossRef\]](#)
7. Holt, C.C. Forecasting seasonals and trends by exponentially weighted moving averages. *Int. J. Forecast.* **2004**, *20*, 5–10. [\[CrossRef\]](#)
8. Jiang, W.; Xu, Y.; Chen, Z.; Zhang, N.; Xue, X.; Liu, J.; Zhou, J. A feature-level degradation measurement method for composite health index construction and trend prediction modeling. *Measurement* **2023**, *206*, 112324. [\[CrossRef\]](#)
9. Lehmann, A. Joint modeling of degradation and failure time data. *J. Stat. Plan. Inference* **2009**, *139*, 1693–1706. [\[CrossRef\]](#)
10. Bellavista, P.; Della Penna, R.; Foschini, L.; Scotece, D. Machine learning for predictive diagnostics at the edge: An IIoT practical example. In Proceedings of the ICC 2020–2020 IEEE International Conference on Communications (ICC), Dublin, Ireland, 7–11 June 2020; pp. 1–7.
11. Li, P.; Pei, Y.; Li, J. A comprehensive survey on design and application of autoencoder in deep learning. *Appl. Soft Comput.* **2023**, *138*, 110176. [\[CrossRef\]](#)
12. Singh, S.A.; Desai, K. Automated surface defect detection framework using machine vision and convolutional neural networks. *J. Intell. Manuf.* **2023**, *34*, 1995–2011. [\[CrossRef\]](#)
13. Tang, T.W.; Hsu, H.; Li, K.M. Industrial anomaly detection with multiscale autoencoder and deep feature extractor-based neural network. *IET Image Process.* **2023**, *17*, 1752–1761. [\[CrossRef\]](#)
14. Pradeep, D.; Vardhan, B.V.; Raiak, S.; Muniraj, I.; Elumalai, K.; Chinnadurai, S. Optimal Predictive Maintenance Technique for Manufacturing Semiconductors using Machine Learning. In Proceedings of the 2023 3rd International Conference on Intelligent Communication and Computational Techniques (ICCT), Jaipur, India, 19–20 January 2023; pp. 1–5.
15. Nuhu, A.A.; Zeeshan, Q.; Safaei, B.; Shahzad, M.A. Machine learning-based techniques for fault diagnosis in the semiconductor manufacturing process: A comparative study. *J. Supercomput.* **2023**, *79*, 2031–2081. [\[CrossRef\]](#)
16. Mao, W.; Feng, W.; Liu, Y.; Zhang, D.; Liang, X. A new deep auto-encoder method with fusing discriminant information for bearing fault diagnosis. *Mech. Syst. Signal Process.* **2021**, *150*, 107233. [\[CrossRef\]](#)

17. Abbasi, S.; Famouri, M.; Shafiee, M.J.; Wong, A. OutlierNets: Highly compact deep autoencoder network architectures for on-device acoustic anomaly detection. *Sensors* **2021**, *21*, 4805. [CrossRef]
18. Yazici, M.T.; Basurra, S.; Gaber, M.M. Edge machine learning: Enabling smart internet of things applications. *Big Data Cogn. Comput.* **2018**, *2*, 26. [CrossRef]
19. Yu, Y.-C.; Chuang, S.-W.; Shuai, H.-H.; Lee, C.-Y. Fast Adaption for Multi Motor Anomaly Detection via Meta Learning and deep unsupervised learning. In Proceedings of the 2022 IEEE 31st International Symposium on Industrial Electronics (ISIE), Anchorage, AK, USA, 1–3 June 2022; pp. 1186–1189.
20. Advantech. WISE-2410—LoRaWAN Wireless Vibration Sensor—Advantech. 2023. Available online: https://www.advantech.com/en/products/b7e2306f-d561-4ca9-b0e3-33f7057e185f/wise-2410/mod_25018dc7-355c-40b4-bf9b-c93f6c73f1a0 (accessed on 1 June 2023.).
21. Advantech. WebAccess_MCM_DS(07.18.17)—Advantech Support—Advantech. 2023. Available online: <https://www.advantech.com/en/support/details/datasheet?id=b5660e1c-d223-40ed-86bd-bdab7be541d7> (accessed on 1 June 2023.).
22. Artono, B.; Susanto, A.; Hidayatullah, N.A. Design of Smart Device for Induction Motor Condition Monitoring. *J. Phys. Conf. Ser.* **2021**, *1845*, 012035. [CrossRef]
23. Lessmeier, C.; Kimotho, J.K.; Zimmer, D.; Sextro, W. Condition monitoring of bearing damage in electromechanical drive systems by using motor current signals of electric motors: A benchmark data set for data-driven classification. In Proceedings of the European Conference of the PHM Society, Bilbao, Spain, 5–8 July 2016; Volume 3.
24. Huang, S.R.; Huang, K.H.; Chao, K.H.; Chiang, W.T. Fault analysis and diagnosis system for induction motors. *Comput. Electr. Eng.* **2016**, *54*, 195–209. [CrossRef]
25. Salem, M.; Taheri, S.; Yuan, J.S. An experimental evaluation of fault diagnosis from imbalanced and incomplete data for smart semiconductor manufacturing. *Big Data Cogn. Comput.* **2018**, *2*, 30. [CrossRef]
26. Ho, T.K. The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 832–844.
27. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282.
28. Dua, D.; Graff, C. *UCI Machine Learning Repository*; University of California, School of Information and Computer Science: Irvine, CA, USA, 2019; Available online: <https://archive.ics.uci.edu/ml/datasets.php> (accessed on 1 June 2023.).
29. Huang, J.; Ling, C.X. Using AUC and accuracy in evaluating learning algorithms. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 299–310. [CrossRef]
30. Type I Error and Type II Error. Available online: <https://explorable.com/type-i-error> (accessed on 12 February 2024).
31. ICML. ICML 2019 Meta-Learning Tutorial. 2019. Available online: <https://sites.google.com/view/icml19metalearning> (accessed on 1 June 2023.).
32. Finn, C.; Abbeel, P.; Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1126–1135.
33. Gardner, E.S., Jr. Exponential smoothing: The state of the art. *J. Forecast.* **1985**, *4*, 1–28. [CrossRef]
34. Ostertagova, E.; Ostertag, O. Forecasting using simple exponential smoothing method. *Acta Electrotech. Et Inform.* **2012**, *12*, 62. [CrossRef]
35. Amo-Salas, M.; López-Fidalgo, J.; Pedregal, D.J. Experimental designs for autoregressive models applied to industrial maintenance. *Reliab. Eng. Syst. Saf.* **2015**, *133*, 87–94. [CrossRef]
36. Zhao, Z.; Liu, F. Industrial monitoring based on moving average PCA and neural network. In Proceedings of the 30th Annual Conference of IEEE Industrial Electronics Society, IECON 2004, Busan, Republic of Korea, 2–6 November 2004; Volume 3, pp. 2168–2171.
37. Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
38. Liang, Y.-H. Combining seasonal time series ARIMA method and neural networks with genetic algorithms for predicting the production value of the mechanical industry in Taiwan. *Neural Comput. Appl.* **2009**, *18*, 833–841. [CrossRef]
39. Han, S.; Pool, J.; Tran, J.; Dally, W. Learning both weights and connections for efficient neural network. *Adv. Neural Inf. Process. Syst.* **2015**, *28*.
40. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2704–2713.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.