electronics

MDPI

Article

Construction of Mining Robot Equipment Fault Prediction Model Based on Deep Learning

Yanshu Li [1,*] and Jiyou Fei [2]

[1] College of Mechanical and Electrical Engineering, Shanxi Datong University, Datong 037009, China
[2] College of Locomotive and Rolling Stock Engineering, Dalian Jiaotong University, Dalian 116028, China; fjy@djtu.edu.cn
* Correspondence: dtulys@sxdtdx.edu.cn

Abstract: In the field of mining robot maintenance, in order to enhance the research on predictive modeling, we introduce the LODS model (long short-term memory network (LSTM) optimized deep fusion neural network (DFNN) with spatiotemporal attention network (STAN)). Traditional models have shortcomings in handling the long-term dependencies of time series data and mining the complexity of spatiotemporal information in the field of mine maintenance. The LODS model integrates the advantages of LSTM, DFNN and STAN, providing a comprehensive method for effective feature extraction and prediction. Through experimental evaluation on multiple data sets, the experimental results show that the LODS model achieves more accurate predictions, compared with traditional models and optimization strategies, and achieves significant reductions in MAE, MAPE, RMSE and MSE of 15.76, 5.59, 2.02 and 11.96, respectively, as well as significant reductions in the number of parameters and computational complexity. It also achieves higher efficiency in terms of the inference time and training time. The LODS model performs well in all the evaluation indexes and has significant advantages; thus, it can provide reliable support for the equipment failure prediction of the mine maintenance robot.

Keywords: deep learning; fault detection; robotic maintenance; predictive modeling; mining industry; data analysis

check for updates

Citation: Li, Y.; Fei, J. Construction of Mining Robot Equipment Fault Prediction Model Based on Deep Learning. *Electronics* **2024**, *13*, 480. https://doi.org/10.3390/electronics13030480

Received: 29 December 2023
Revised: 18 January 2024
Accepted: 19 January 2024
Published: 23 January 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

In the field of industrial automation, intelligent mine maintenance robots are playing an increasingly critical role. Faced with the extreme characteristics of the mining environment and the high complexity of maintaining equipment, they must cope with the sudden and unpredictable nature of equipment failures [1]. The characteristics of equipment failures increase the difficulty of maintenance work, forcing us to consider developing advanced predictive models to effectively address them.

Deep learning, as a powerful data analysis and predictive modeling technology, has demonstrated its outstanding capabilities in multiple fields. In particular, its performance in feature learning and pattern recognition has proven to be beneficial in mining potential information in data, thereby improving the accuracy and efficiency of prediction [2]. However, there are several challenges that need to be overcome to successfully apply deep learning technology to the predictive maintenance of mine maintenance robots. The data generated by these robots often have complex multi-modal timing characteristics, including but not limited to sensor data, images and sounds [3]. The traditional method that uses single-modal data as the main body of analysis fails to comprehensively examine and understand the operating status of the equipment, making it difficult to meet the needs of practical applications [4].

With the aid of deep learning technology, we expect to overcome the limitations of current methods and achieve more accurate and reliable equipment failure predictions. The high adaptability and autonomous learning capabilities of deep learning make it a

powerful tool to solve multi-modal time series data problems [5]. By further improving and adapting the deep learning model, we expect to achieve a qualitative leap forward in the field of fault prediction for mine maintenance robots.

Most current research focuses on the processing of single-modal data and ignores the importance of the comprehensive analysis of multi-modal data. At the same time, the characteristics of long-term dependencies in time series data and the complexity of spatiotemporal data pose additional challenges to predictive modeling. This study will provide a comprehensive evaluation of several currently widely used deep learning models, aiming to explore their potential application scenarios and challenges in the field of mine robot maintenance and provide a theoretical basis for the model construction in this study.

As a powerful image processing tool in deep learning, the convolutional neural network (CNN) can effectively capture the spatial hierarchical features of images through its stacked convolutional layers and pooling layers [6]. This means that the CNN shows great potential in image analysis tasks in the field of mining robot maintenance, especially in identifying tiny cracks, corrosion or other forms of defects on the surfaces of equipment [7]. However, in the face of analysis tasks involving complex multi-modal time series data, such as processing images, sound and sensor data simultaneously, the CNN needs to integrate additional structures for effective information fusion [8].

The temporal convolutional network (TCN) adopts a one-dimensional convolution structure [9]. Its main advantage is to effectively learn the local features of temporal data and it has excellent short-term dependency capture capabilities [10]. The TCN has great application potential in mine maintenance diagnosis. It can analyze time series data generated by sensors to monitor and predict equipment performance [11]. However, the TCN may not be effective in capturing long-term dependencies.

Graph convolutional networks (GCNs) are specifically designed to process graph data and are capable of identifying complex patterns in graph structures [12]. In mining robot maintenance, GCNs can reveal the interactions and dependencies between equipment, providing a more comprehensive system perspective for prediction [13]. However, GCNs face challenges in terms of computational efficiency and scalability when facing large device networks [14].

The Transformer model, with its self-attention mechanism as its core, provides a new solution for the capturing of long-distance dependencies of sequences [15]. When applied to mining robot maintenance, the Transformer model is able to process time series data from different modalities and provide accurate predictions by capturing inherent global dependencies [16]. However, Transformer's high computing requirements limit its deployment in resource-constrained practical application scenarios [17].

Although deep learning models such as CNN, TCN, GCN and Transformer have achieved certain results in the field of mine robot maintenance, they still have shortcomings in processing multi-modal time series data and long-term dependencies in special environments. Traditional CNNs may face difficulties in information fusion [18], The TCN may have certain challenges in modeling long-term dependencies [19], the GCN may have computational efficiency issues in large-scale graph data processing [20], and Transformer may face some computational complexity issues [21].

Based on the shortcomings of the above methods, we propose the LODS model (LSTM-optimized DFNN with STAN). The LODS model uses the long short-term memory network (LSTM) as an optimization module and integrates a deep fusion neural network (DFNN) and spatiotemporal attention network (STAN) through synergy. The DFNN is responsible for processing multi-modal data, while the STAN focuses on the characteristics of spatiotemporal data. The LSTM optimization module helps to integrate the two aspects of information and is especially suitable for processing time series data, improving the accuracy of the model's time series prediction of equipment failures. The LODS model provides a comprehensive and efficient deep learning solution for the field of mining robot maintenance, enabling us to more comprehensively understand the equipment status and improve the accuracy and reliability of fault prediction. Its advantages are

reflected in the comprehensive analysis of multi-modal time series data, spatiotemporal data modeling and long-term dependency processing, providing more reliable decision support for mine maintenance work. By introducing the LSTM optimization module, the model achieves better information collaboration between different modalities and data characteristics, providing an innovative solution for the equipment failure prediction of mine maintenance robots.

The main contributions of this study are as follows:

- The pioneering nature of the LODS model: This study proposes the LODS model, which is an innovative multi-modal time series data analysis framework that combines a deep fusion neural network (DFNN), spatiotemporal attention network (STAN) and long short-term memory network (LSTM). It provides a new modeling method for the field of mine robot maintenance.
- Multi-modal data fusion: The LODS model uses DFNN to deeply integrate various data sources, which enhances the model's understanding of the equipment status and provides more comprehensive information support for fault prevention.
- Spatiotemporal dependency optimization: The STAN module is used to enhance the capture of key spatiotemporal data, and the LSTM module is used to improve the long-term dependency modeling capabilities, greatly improving the accuracy of equipment failure prediction.

In the following sections, we will structure our discussion as follows. Section 2 will introduce the method in depth and reveal the core construction and design principles of the LODS model. Section 3 will focus on the experimental settings and details in order to reproduce the experiment. Section 4 will introduce the experimental results in detail to show the performance of the LODS model in different data sets and scenarios. Finally, Section 5 discusses and summarizes the full text.

2. Methods

2.1. Overview of Our Network

The LODS model (LSTM-optimized DFNN with STAN) that we propose aims to provide a comprehensive and efficient deep learning solution for mining robot equipment failure prediction. The core construction of the LODS model includes three key components: DFNN, STAN and LSTM.

The DFNN is responsible for processing multi-modal data, including sensor data, images and sounds, etc. Through the deep fusion layer, the DFNN effectively fuses the information processed in different branches to achieve a comprehensive understanding of the multi-modal information. The purpose of this section is to provide more comprehensive device information and promote a comprehensive understanding of the device's health status. The STAN module uses an attention mechanism to focus on the spatiotemporal data characteristics in the mining environment. This helps the network to better focus on critical moments and areas and increase its sensitivity to changes in the device status. The role of the STAN in the mine maintenance environment is to capture the complexity of the spatiotemporal data and enhance the network's understanding of the entire system status. The LSTM serves as an optimization module and integrates the output of the DFNN and STAN. It is particularly useful in capturing long-term dependencies in time series data. Through the introduction of the LSTM, we optimize the model's timing prediction accuracy for equipment failures and improve the performance of the overall model.

The network construction process of the LODS model is divided into the following steps. First, we design the DFNN network branch to process multi-modal data, including images, sensor data and sounds. Each branch contains an adaptive hierarchy to extract features of a specific data type. Then, the STAN network branch is designed, which is used to process spatiotemporal data and uses the attention mechanism to capture information at key moments and regions. This branch models multi-modal data in the spatiotemporal dimension. Then, we perform information fusion, using the deep fusion layer to fuse the information of the DFNN and STAN branches to form a global, multi-modal feature

representation. Finally, the LSTM is introduced as an optimization module to integrate the fused feature representation to better capture the long-term dependencies in the time series data and improve the time series prediction performance of the model.

The overall model structure diagram is shown in Figure 1.

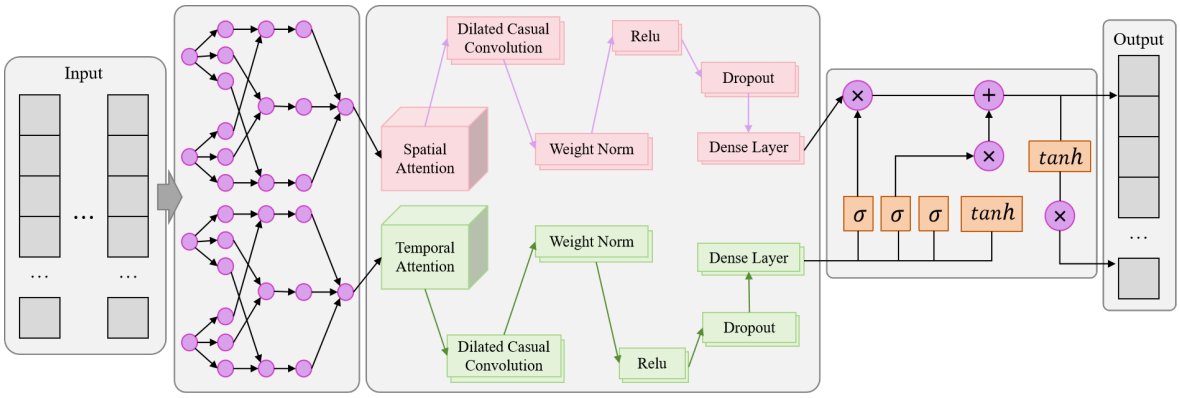

Figure 1. Overall flow chart of the model.

The running process of the LODS model is shown in Algorithm 1.

Algorithm 1: LODS Training

Input: Training data $\mathcal{D}$ from C-MAPSS, SECOM, PHM08, Tennessee Eastman
Output: Trained parameters of the LODS model
Initialize LSTM parameters θ_{LSTM}, DFNN parameters θ_{DFNN}, STAN parameters θ_{STAN}
Initialize optimizer parameters $\theta_{\text{optimizer}}$
for each training epoch **do**
  for each batch in $\mathcal{D}$ **do**
    Sample a batch of multi-modal time series data
    Extract features using DFNN: $F_{\text{DFNN}} = \text{DFNN}(X; \theta_{\text{DFNN}})$
    Extract spatiotemporal features using STAN: $F_{\text{STAN}} = \text{STAN}(X; \theta_{\text{STAN}})$
    Concatenate DFNN and STAN features: $F_{\text{concat}} = \text{concat}(F_{\text{DFNN}}, F_{\text{STAN}})$
    Apply LSTM for sequence modeling: $Y_{\text{pred}} = \text{LSTM}(F_{\text{concat}}; \theta_{\text{LSTM}})$
    Compute loss: $\mathcal{L} = \text{MSE}(Y_{\text{pred}}, Y_{\text{true}})$
    Backpropagate gradients and update LSTM, DFNN, STAN parameters
    Update optimizer parameters using backpropagation: $\theta_{\text{optimizer}} = \text{Adam}(\nabla \mathcal{L}; \theta_{\text{optimizer}})$
  end for
end for
return Trained parameters: $\theta_{\text{LSTM}}, \theta_{\text{DFNN}}, \theta_{\text{STAN}}$

In this model, for the "Input" section, the input data of our model include three main types: one is the sensor data, which are the time sequence data from different sensors of mine robots and may include the numerical sequences of temperature, pressure, vibration and other indicators; the other is image data, which refer to the images of the mine robot's working environment or continuous camera frame sequences, used to capture the visual state information of the device; the other is sound data for the acoustic signals captured during the robot's operation, which can help to detect the abnormal noise of the device to assist in fault prediction. The LODS model uses the DFNN network to process the above multi-modal data and performs deep fusion. For the "Output" section, the data type output by the model is the prediction of the potential failure of the mine robot equipment. Specifically, the output data can be qualitatively defined as probability values or classification results, meaning that every possible failure type will be assigned a predicted

probability by our model. Further, these prediction results can help the maintenance team to make timely repair or maintenance decisions.

2.2. Deep Fusion Neural Network

The deep fusion neural network (DFNN) is a deep learning model focused on processing multi-modal data. The basic principle is to process different types of data, such as sensor data, images and sounds, in different branches of the network, and then effectively fuse the information of these branches through a deep fusion layer to improve the prediction performance for equipment failures [22]. The main purpose of the DFNN is to achieve the organic integration of different data types in complex multi-modal data scenarios to more comprehensively understand the status of the system [23].

In the application of mine robot maintenance, the DFNN model shows its powerful comprehensive data processing capability, especially in processing multi-modal data. Unlike traditional methods, the DFNN model not only deals with a single form of data, such as the readout of the sensor, but also can integrate and analyze multi-modal data from different sources, including but not limited to high-resolution images, complex sound signals and multi-dimensional sensor data. For example, when monitoring the health of key maintenance components such as hydraulic arms, the DFNN can use data from vibration sensors, acoustic monitoring equipment and visual detection systems, which are subtle signals that may be missed by a single data source.

The internal architecture of the DFNN is specifically designed with a deep fusion layer, which is tailored to the characteristics of different data modalities and the optimal way to extract features. For example, our model uses convolutional neural network branches to process image data to identify visual patterns and texture abnormalities, while using recurrent neural network branches to process sound signals to capture temporal dependence and periodicity characteristics. Using the features obtained by these branches, the deep fusion layer of the DFNN uses innovative fusion algorithms to effectively combine them and establish meaningful associations between different data streams.

In the LODS model constructed in this work, the DFNN undertakes the task of processing multi-modal data. Its function is to organically integrate information from different branches through a deep fusion layer to form a global feature representation. This fusion of multi-modal information enables the model to more comprehensively understand the working status of the equipment, providing a more accurate basis for the prediction of equipment failures. The function of the DFNN in the overall model is to provide the efficient processing of different data types and generate more informative feature representations through fusion, thereby providing strong support for subsequent spatiotemporal data modeling and the optimization of long-term dependencies. Its importance lies in its ability to provide global and multi-dimensional device information for the LODS model, making a key contribution to improving the performance of the overall model.

The structure diagram of the DFNN model is shown in Figure 2.

The main formula of the DFNN is as follows:

$$Z_t = \sigma(\mathbf{W}_z * X_t + \mathbf{U}_z * H_{t-1} + \mathbf{b}_z) \tag{1}$$

where Z_t is the update gate at time t, σ is the sigmoid activation function, $\mathbf{W}_z$ and $\mathbf{U}_z$ are weight matrices for the input and hidden state, X_t is the input at time t, H_{t-1} is the hidden state at time $t-1$, and $\mathbf{b}_z$ is the bias vector for the update gate.

$$R_t = \sigma(\mathbf{W}_r * X_t + \mathbf{U}_r * H_{t-1} + \mathbf{b}_r) \tag{2}$$

where R_t is the reset gate at time t, σ is the sigmoid activation function, $\mathbf{W}_r$ and $\mathbf{U}_r$ are weight matrices for the input and hidden state, X_t is the input at time t, H_{t-1} is the hidden state at time $t-1$, and $\mathbf{b}_r$ is the bias vector for the reset gate.

$$\tilde{H}_t = \tanh(\mathbf{W}_h * X_t + \mathbf{U}_h * (R_t \odot H_{t-1}) + \mathbf{b}_h) \tag{3}$$

where $\tilde{H}_t$ is the candidate hidden state at time t, tanh is the hyperbolic tangent activation function, $\mathbf{W}_h$ and $\mathbf{U}_h$ are weight matrices for the input and hidden state, X_t is the input at time t, R_t is the reset gate at time t, H_{t-1} is the hidden state at time $t-1$, and $\mathbf{b}_h$ is the bias vector for the candidate hidden state.

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t \tag{4}$$

where H_t is the hidden state at time t, Z_t is the update gate at time t, and $\odot$ denotes element-wise multiplication.

$$\hat{Y}_t = \text{argmax}(Y_t) \tag{5}$$

where $\hat{Y}_t$ is the predicted class at time t, and *argmax* is the argmax function that outputs the index of the maximum element.

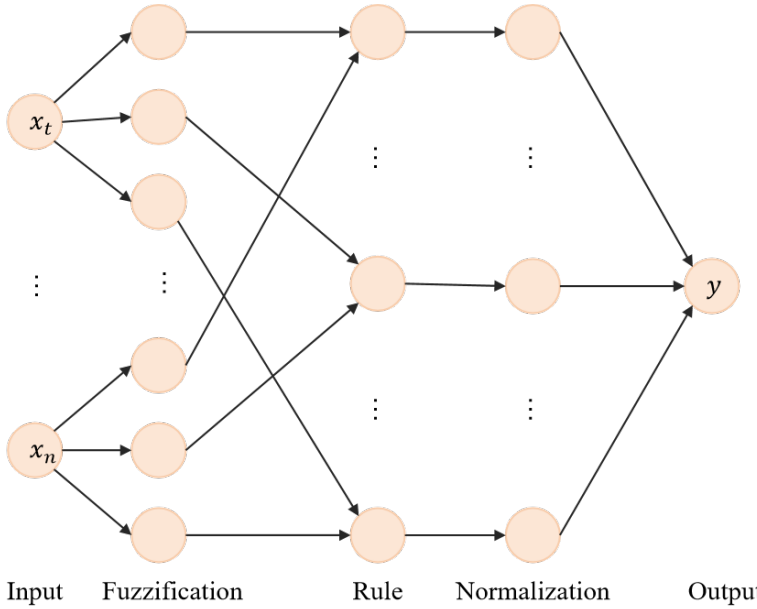

Figure 2. Flow chart of the DFNN model.

2.3. Spatiotemporal Attention Network

The spatiotemporal attention network (STAN) is a deep learning model focused on processing spatiotemporal data [24]. The basic principle is to use the attention mechanism to enable the network to better focus on key moments and areas, thereby enhancing the modeling ability for spatiotemporal data [25]. The main purpose of the STAN is to capture temporal and spatial relationships in the data and it is suitable for data scenarios where time and space characteristics need to be considered.

In the practice of mine robot maintenance, there are complex time sequence data and location correlations. By introducing an attention mechanism, this model not only accurately calibrates key time points, but also identifies areas with a significant impact on the performance of the device. The STAN model is significantly better in capturing the intrinsic connections between spatiotemporal data than traditional deep learning methods. Through the built-in spatial and temporal attention component, the model can identify the patterns that have a significant impact on the running state of the entire mine machine in the continuous data flow. For example, when predicting conveyor belt wear, the STAN uses its spatiotemporal attention mechanism to pinpoint abnormal wear patterns occurring in specific operating conditions, thus providing decision support for early maintenance and replacement. The STAN model utilizes the attention scoring mechanism to prioritize the data regions that are most important for the current prediction target. In practical cases, when analyzing noise data generated by multiple sensors, it can distinguish abnormal

sounds generated by equipment failures and distinguish them from background ambient noise, thus improving the fault detection rate and reducing false positives. The flexibility of the STAN is also reflected in its ability to process spatiotemporal data in different mining environments and equipment. For example, when monitoring the navigation system of a mine robot, the STAN is able to adapt to different terrain and obstacle configurations and optimize the path planning by analyzing the evolution of spatiotemporal data.

In the LODS model, the function of the STAN module is to model spatiotemporal data and enhance the understanding of the entire system state. Due to the complexity of spatiotemporal relationships in the mining environment, the introduction of STAN helps the network to better focus on critical moments and areas and improves its sensitivity to changes in equipment status. Its importance lies in the ability to provide the overall model with effective modeling capabilities for spatiotemporal data, thereby providing more accurate and comprehensive support for the prediction of equipment failures. In the equipment failure prediction of mine maintenance robots, the contribution of the STAN module not only improves the performance of the model but also strengthens the ability to adapt to the complexity of the mine environment.

The structure diagram of the STAN model is shown in Figure 3.

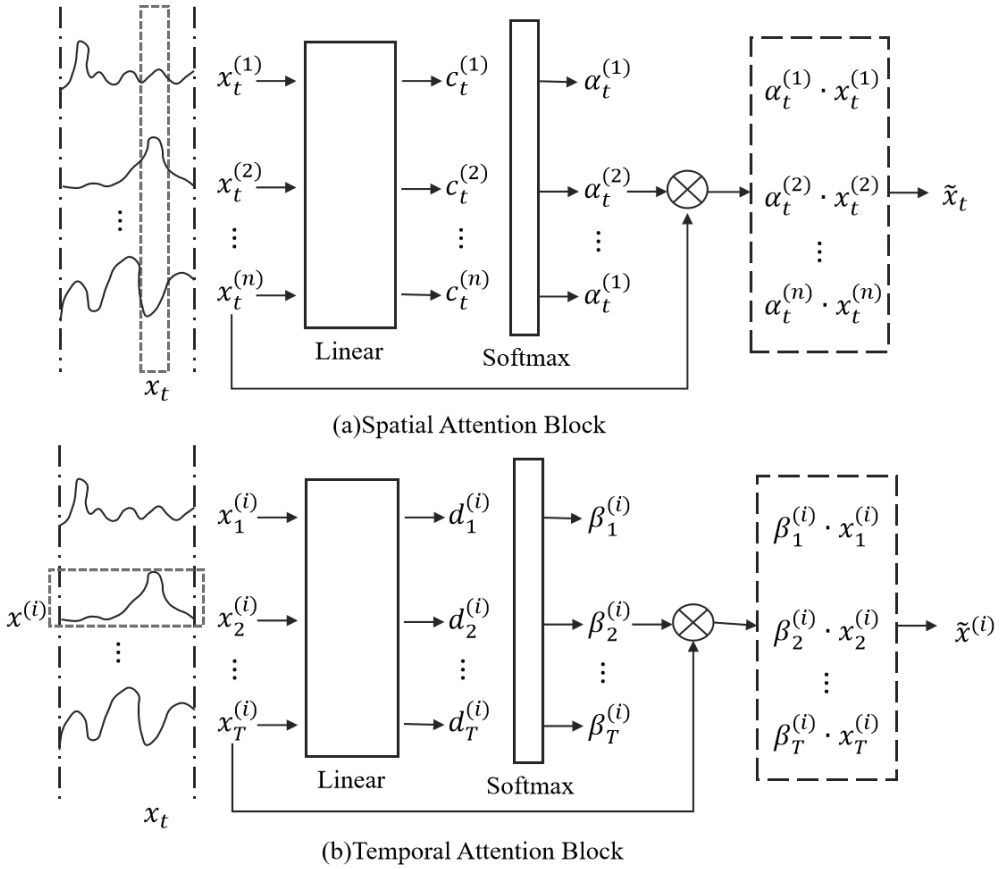

Figure 3. Flow chart of the spatiotemporal attention network model.

The main formula of the STAN model is as follows:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \tag{6}$$

where Q is the query matrix, K is the key matrix, V is the value matrix, X is the input matrix, and W_Q, W_K, W_V are learnable weight matrices.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

where $Attention$ is the attention mechanism, $softmax$ is the softmax function, and d_k is the dimension of the key vectors.

$$MultiHead(X) = Concat(head_1, \ldots, head_h)W_O \tag{8}$$

where $MultiHead$ is the multi-head attention, $head_i = Attention(XW_{Qi}, XW_{Ki}, XW_{Vi})$, $i = 1, \ldots, h$, and W_O is the output weight matrix.

$$Residual(X) = X + SubLayer(X) \tag{9}$$

where $Residual$ is the residual connection. $SubLayer$ is any of the sub-layers.

2.4. Long Short-Term Memory

The long short-term memory network (LSTM) is a variant of the recurrent neural network (RNN) with long short-term memory capabilities. The basic principle is that by introducing gate structures (forgetting gates, input gates and output gates), the LSTM model can better capture the long-term dependencies in time series data and effectively avoid the vanishing gradient problem [26]. The main use of LSTM is to process time series data, especially in scenarios where long-term dependencies need to be captured, such as speech recognition, text generation, etc. [27].

LSTM demonstrates its profound value in mine robot maintenance tasks, especially when processing and analyzing time series data collected by monitoring equipment. Mine robots usually operate in extreme and unpredictable environments; the data collected reflect the operating state of the robot's systems, and these data show complex temporal characteristics. The LSTM model is suitable for such application scenarios, firstly, because the gating mechanism of LSTM is particularly suitable for analyzing the time series recorded by robotic sensors. In the mine environment, robotic components (e.g., drill bits, loading arms, delivery systems) may gradually experience subtle performance degradation, which show long-term trends in temporal data. By identifying these long-term dependencies, the LSTM can alert them to potential failures early, leaving more time for maintenance teams to respond. Second, in practice, LSTM has significant advantages over traditional RNN models in handling the stability and accuracy of long time series. This advantage is attributed to the gate structure in its design, which can effectively prevent the gradient disappearance or explosion problem and ensure the training stability. For mining robots, this means that the model can be used to track and analyze changes in its key operational parameters, allowing an accurate assessment of the robot's long-term health. In addition, due to the inherent timing characteristics of the mine robot maintenance data, the LSTM model provides a more accurate and reliable prediction tool. By modeling the timing changes in device status, LSTM helps to develop maintenance plans and preventive maintenance strategies, reduce unexpected downtime and extend the operating life of the robot.

In the LODS model, the function of the LSTM module as an optimization module is to integrate the output of the DFNN and STAN branches to enhance the overall model's ability to process time series data. By introducing LSTM, we can better capture the long-term dependencies of equipment state changes and improve the accuracy of the model in time series prediction. Its importance lies in providing the overall model with optimized support for time series data. By effectively integrating information from different branches, the model's sensitivity and expressive ability regarding changes in equipment failure time series are improved. In the field of mining robot maintenance, the introduction of the LSTM module strengthens the model's performance in processing equipment failure time series data, providing important support for the reliability of the entire prediction model.

The structure diagram of the LSTM model is shown in Figure 4.

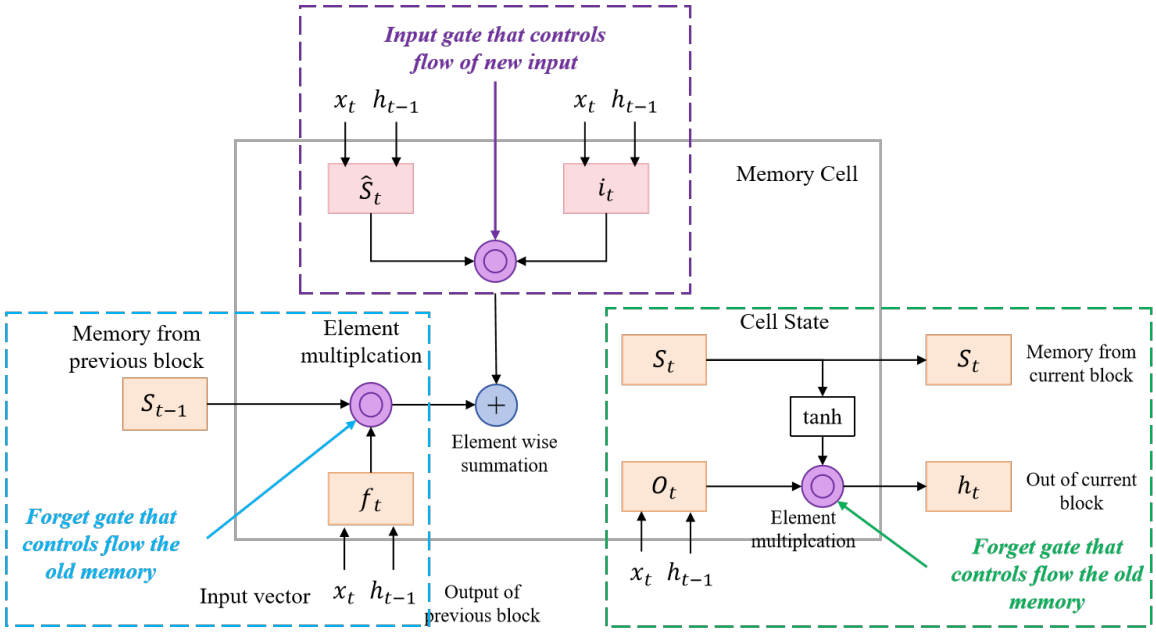

Figure 4. Flow chart of the LSTM model.

The main formula of the attention mechanism is as follows:

$$ForgetGate: f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{10}$$

where f_t is the forget gate output at time t, W_f is the weight matrix for the forget gate, h_{t-1} is the hidden state from the previous time step, x_t is the input at time t, b_f is the bias vector for the forget gate, and σ is the sigmoid activation function.

$$InputGate: i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{11}$$

where i_t is the input gate output at time t, W_i is the weight matrix for the input gate, and b_i is the bias vector for the input gate.

$$CellCandidate: \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{12}$$

where $\tilde{C}_t$ is the cell candidate at time t, W_C is the weight matrix for the cell candidate, b_C is the bias vector for the cell candidate, and tanh is the hyperbolic tangent activation function.

$$CellState: C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{13}$$

where C_t is the cell state at time t, and $\odot$ is the element-wise multiplication (Hadamard product).

$$OutputGate: o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{14}$$

where o_t is the output gate output at time t, W_o is the weight matrix for the output gate, and b_o is the bias vector for the output gate.

$$HiddenState: h_t = o_t \odot \tanh(C_t) \tag{15}$$

where h_t is the hidden state at time t.

3. Experiment

3.1. Experimental Environment

- Hardware Environment. The hardware environment used in the experiments consists of a high-performance computing server equipped with an AMD Ryzen Threadripper 3990X @ 3.70 GHz CPU and 1 TB RAM, along with 6 Nvidia GeForce RTX 3090 24 GB GPUs. This remarkable hardware configuration provides outstanding computational and storage capabilities for the experiments, especially well suited for training and

inference tasks in deep learning. It effectively accelerates the model training process, ensuring efficient experimentation and rapid convergence.

- Software Environment. In this study, we utilized Python and PyTorch to implement our research work. Python, serving as the primary programming language, provided us with a flexible development environment. PyTorch, as the main deep learning framework, offered powerful tools for model construction and training. Leveraging PyTorch's computational capabilities and automatic differentiation functionality, we were able to efficiently develop, optimize, and train our models, thereby achieving better results in the experiments.

3.2. Experimental Data Sets

In order to obtain comprehensive and representative experimental results, we selected four different data sets, including the C-MAPSS data set, SECOM Manufacturing Data, PHM08 Challenge Data and Tennessee Eastman Process Simulation Data.

The C-MAPSS Data Set is a classic data set used for aviation engine health monitoring, including multiple different types of engines. Each engine is equipped with multiple sensors, providing rich timing data [28]. This data set is widely used in the aviation field, providing complex and diverse engine operating states for the model and helping to verify the performance of the LODS model in complex environments. In this study, we selected engine data similar to the mine equipment sensor configuration to meet the fault prediction requirements of the LODS model in the mine-specific environment.

SECOM Manufacturing Data comes from the semiconductor manufacturing industry. It contains a range of sensor data from the semiconductor production process, covering both functioning and defective products [29]. The main characteristic of SECOM Manufacturing Data lies in its ability to capture diverse issues in the manufacturing process, providing the LODS model with training data that are closer to actual application scenarios. In this experiment, the features related to the fault detection in the production process of the mine robot were extracted, and the model was adjusted to accurately identify the unique fault signal of the mine robot.

PHM08 Challenge Data is a data set from the PHM community, specifically designed for mechanical system health prediction [30]. This data set contains sensor data from multiple devices and is complex and challenging. The use of PHM08 Challenge Data aims to test the LODS model's ability to process the multi-source data of mechanical systems and its performance in challenging environments. This data set is used to evaluate and optimize the prediction accuracy of the LODS model under high load conditions by simulating the high load during the mine machine operation and its impact on the health state of the equipment.

Tennessee Eastman Process Simulation Data is an extensively researched chemical process simulation data set. This data set simulates the operation of a chemical plant and contains simulated sensor data under normal and fault operating conditions [31]. By using this data set, we aim to verify the robustness and reliability of the LODS model in handling complex failure prediction tasks in chemical processes. This data set is used to simulate the chemical factors that may affect the health of the robot equipment in the mine environment, and to test and verify the failure prediction model for this special scenario.

3.3. Experimental Setup and Details

In this paper, the LODS model (LSTM-optimized DFNN with STAN) model is constructed to meet the challenge of the fault prediction of mine maintenance robot equipment. To ensure the accuracy and reproducibility of the experiment, by thoroughly considering the design and details of the experiment, the specific experimental settings are detailed below.

Step 1: Data set preparation

- Data cleaning: In this paper, the features with more than 5% missing values are set as the objects to be processed, and the mean filling method is used for processing.

- Data standardization: This paper adopts the standard deviation standardization method to map the values of all features to the standard normal distribution with mean 0 and standard deviation 1.
- Data split: 70% of the data in the experiment are used for training, 20% for validation and 10% for testing. Random sampling is used to ensure the stochasticity and representativeness of the data set.

Step 2: Model training

- Network parameter setting: The initial learning rate is 0.001, the batch size is 64, and 150 rounds of iterative training are conducted.
- Model architecture design: The DFNN is responsible for processing multi-modal data, the STAN focuses on spatial and temporal relations and LSTM serves as an optimization module to integrate the output of both. Three branches are included in the DFNN that process sensor data, images and sound information. The STAN uses a three-layer attentional structure to model the relevance of data in spatiotemporal dimensions. The number of hidden cells of the LSTM was set to 128 to ensure that it effectively captured the long-term dependencies in the temporal data.
- Model training process: A stochastic gradient descent (SGD) optimizer was used, combined with a learning rate decay mechanism. During the training process, training was stopped to avoid overfitting when 10 consecutive rounds of validation set loss no longer decreased. Moreover, batch normalization and dropout were used to improve the robustness and generalization of the model.

Step 3: Model validation and tuning

- Cross-validation: The experiment uses k-fold cross-validation to divide the data set into k mutually exclusive subsets. Moreover, $k = 10$, to balance the computational cost and the accuracy of the performance evaluation.
- Model fine-tuning: During the fine-tuning process, we focus on the model hyperparameters such as the learning rate and batch size, as well as the details of the network structure. By systematically using different hyperparameter combinations, we chose the parameter settings to achieve the best performance on the validation set. To prevent overfitting, we also used regularization techniques to limit the complexity of the model.

Step 4: Ablation experiment

During the experiments in this paper, a series of ablation experiments were designed to investigate the impact of the various components of the LODS model on model performance. Remove experiments for each component were performed to assess its contribution to the overall model performance.

- Removing LSTM: This set of experiments removed the LSTM components from the overall LODS model and built the model using only the DFNN and STAN. The number of hidden cells in LSTM was set to 50 with a learning rate of 0.001.
- Remove STAN: This set of experiments removed STAN components from LODS and modeled only using the DFNN and LSTM. The number of attention heads in STAN was 4 with a learning rate of 0.002.
- Removing DFNN: This set of experiments retained the LSTM and STAN and removed the DFNN. The number of convolution kernels for DFNN was set to 16, with a learning rate of 0.001.
- Overall model LODS: In the full model, the LSTM, STAN and DFNN work together to comprehensively utilize temporal data, spatial–temporal relationships and multi-modal information. The number of hidden cells in LSTM is 100, the number of the heads of attention in STAN is 8, and the number of convolution kernels in the DFNN is 32, with an overall learning rate of 0.001.

Step 5: Comparative Analysis

In the experiments presented in this paper, comparison experiments were also designed, focusing on the optimization strategy, selecting Adam, Bayesian and PSO as comparison objects, and performing the performance comparison with LSTM.

- Adam vs. LSTM: In this comparison experiment, the learning rate of Adam was 0.001, and the number of hidden cells of LSTM was 100.
- Bayesian vs. LSTM: In this comparison experiment, the default settings of the Bayesian optimization strategy were adopted, and the number of hidden cells of LSTM was set to 100.
- PSO vs. LSTM: In this comparison experiment, the number of iterations of PSO was set to 50, the population size was set to 20, and the number of hidden cells of LSTM was set to 100.

Step 6: Model Evaluation In this study, we conducted a comprehensive and detailed evaluation of the LODS model, focusing on the two core dimensions of accuracy and efficiency.

- The accuracy of the model's prediction ability is assessed using the following evaluation metrics: Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE) and Mean Square Error (MSE). A lower MAE value indicates greater accuracy in the model's predictions. A low MAPE value indicates small errors and high prediction accuracy. RMSE permits a more stringent evaluation of the accuracy of the model by giving greater weight to larger errors. MSE is a key indicator in evaluating the model prediction performance in regression analysis.
- The efficiency of the model is evaluated using the following indicators: parameters, floating-point operations (FLOPs), inference time and training time. Models with fewer parameters are generally considered more efficient and concise. Lower FLOPs means that the model has a smaller demand for computing resources and has higher computing efficiency. The inference time directly reflects the real-time nature and response speed of the model. Training time measures the total time that it takes the model to complete the learning process during the training phase.

4. Results

We performed a comprehensive comparison of the LODS model against six alternative models across four distinct data sets. The evaluation incorporated several metrics, including the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Root Mean Square Error (RMSE) and Mean Square Error (MSE), alongside additional indicators, as detailed in Table 1. On the C-MAPSS data set, compared with other models, the LODS model achieved significant reductions in MAE, MAPE, RMSE and MSE, which were 15.76, 5.59, 2.02 and 11.96, respectively. This shows that the LODS model predicts equipment failures more accurately on the C-MAPSS data set. On the three data sets of SECOM Manufacturing Data, PHM08 Challenge Data and Tennessee Eastman Process Simulation Data, the LODS model also performed well. Compared with other models, it achieved lower MAE, MAPE, RMSE and MSE values, further proving its superiority on multiple data sets. It is particularly worth noting that, compared with the classic model mentioned earlier, the LODS model has achieved significant improvements in various indicators. This shows that the LODS model can better adapt to the complex multi-modal time series data of mine maintenance robots and improve the accuracy and reliability of equipment failure prediction. Comparing all data results, the LODS model performs well in all evaluation indicators and has significant advantages in predicting equipment failure in mine maintenance robots.

Figure 5 visually presents the excellent performance of the LODS model in various performance indicators. It can be clearly observed from the figure that the LODS model has significant advantages over other models in indicators such as MAE, MAPE, RMSE and MSE.

Table 1. Performance comparison across models Using varied data sets.

Data Set	Model	MAE	MAPE	RMSE	MSE
C-MAPSS Data	Wen [32]	45.31	12.62	5.63	20.83
	Long [33]	37.69	11.43	8.01	12.45
	Wang [34]	22.3	11.08	4.46	15.03
	Chen [35]	22.98	11.82	5.82	12.98
	Pinto [36]	40.08	13.37	5.65	21.59
	Shaheen [37]	38.92	11.84	4.76	20.87
	Ours (LODS)	15.76	5.59	2.02	11.96
SECOM Data	Wen	33.75	11.66	7.37	14.42
	Long	48.45	10.37	8.11	20.46
	Wang	49.32	9.86	4.31	16.98
	Chen	26.98	12.31	7.78	30.39
	Pinto	50.23	12.22	4.42	18.58
	Shaheen	49.22	14.98	4.77	14.93
	Ours (LODS)	17.74	4.8	2.3	12.2
PHM08 Data	Wen	22.42	10.1	5.16	27.92
	Long	49.89	8.77	7.33	12.37
	Wang	36.03	9.82	7.34	21.47
	Chen	21.54	11.41	4.66	20.75
	Pinto	21.03	10.56	6.56	26.69
	Shaheen	42.29	14.56	4.84	27.33
	Ours (LODS)	15.97	4.62	2.69	10.97
TEPS Data	Wen	29.3	14.14	5.03	21
	Long	41.9	12.32	5.51	20.18
	Wang	37.48	13.87	7.32	13.98
	Chen	24.02	10.22	5.05	26.74
	Pinto	35.29	15.29	7.6	25.41
	Shaheen	26.35	12.34	5.39	29.61
	Ours (LODS)	16.02	8.25	4.11	11.98

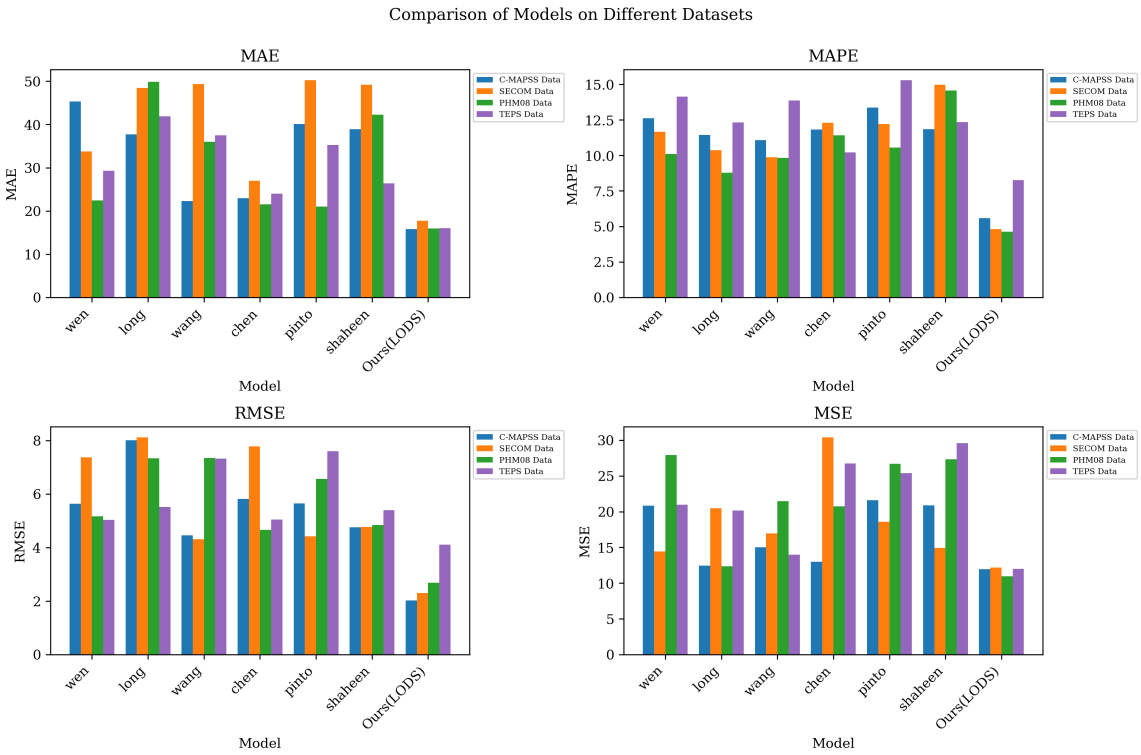

Figure 5. Model accuracy verification comparison chart of different indicators of different models.

As shown in Table 2, we compared the LODS model with six other models on four different data sets, using metrics such as the model parameter volume, computational complexity (FLOPs), inference time and training time to evaluate the model performance.

On the C-MAPSS data set, the parameter amount, computational complexity, inference time and training time of the LODS model are 337.64 M, 3.54 G, 5.37 ms and 325.19 s, respectively. Compared with other models, the LODS model has achieved significant reductions in parameter volume and computational complexity, while also achieving higher efficiency in terms of inference time and training time. The LODS model also performed well on the three data sets of SECOM Manufacturing Data, PHM08 Challenge Data and Tennessee Eastman Process Simulation Data. Specifically, the number of parameters, computational complexity, inference time and training time of the LODS model on these three data sets are significantly better than those of other models, further proving its efficiency on multiple data sets. Of particular note is that the LODS model performs well in terms of computational complexity and has lower FLOPs than other models, indicating that it improves the prediction performance while reducing the computational costs. Overall, the LODS model performed well in all evaluation indicators, not only achieving significant improvements in prediction performance, but also achieving higher efficiency in terms of model parameters and computational complexity.

Table 2. Model efficiency verification and comparison of different indicators of different data sets.

Data Set	Model	Parameters (M)	FLOPs (G)	Inference Time (ms)	Training Time (s)
C-MAPSS Data	Wen	526.37	5.36	7.32	540.79
	Long	642.52	5.64	8.02	757.03
	Wang	385.58	5.85	7.56	674.11
	Chen	757.32	5.69	7.34	738.50
	Pinto	416.64	5.89	7.91	394.13
	Shaheen	530.76	5.80	8.33	523.55
	Ours (LODS)	337.64	3.54	5.37	325.19
SECOM Data	Wen	541.95	5.83	9.72	601.75
	Long	788.92	6.92	12.19	809.85
	Wang	451.04	7.39	10.79	472.44
	Chen	746.65	7.42	11.08	690.99
	Pinto	410.04	4.48	8.15	458.64
	Shaheen	447.23	5.58	8.38	488.27
	Ours (LODS)	317.86	3.64	5.60	336.74
DPHM08 Data	Wen	580.73	5.64	7.86	476.10
	Long	791.76	6.73	13.29	626.74
	Wang	797.16	4.36	6.44	695.68
	Chen	645.90	7.83	10.77	683.59
	Pinto	498.69	4.73	6.88	406.17
	Shaheen	504.77	5.80	8.69	466.95
	Ours (LODS)	337.64	3.53	5.32	326.77
TEPS Data	Wen	498.05	5.20	9.96	600.21
	Long	639.90	7.84	13.99	841.01
	Wang	745.59	6.50	7.99	666.11
	Chen	659.58	6.57	10.22	667.46
	Pinto	431.99	5.07	7.38	504.74
	Shaheen	471.67	5.85	8.87	565.30
	Ours (LODS)	317.41	3.63	5.63	338.30

Figure 6 further visually presents the performance comparison of the LODS model on different data sets. By observing the chart, it can be clearly found that the LODS model performs well on multiple data sets, such as C-MAPSS, SECOM Manufacturing Data, PHM08 Challenge Data and Tennessee Eastman Process Simulation Data, demonstrating its performance in different mine maintenance robot task scenarios. It shows universality and robustness.

As shown in Table 3, we conducted a series of ablation experiments to remove the LSTM, STAN and DFNN components in the LODS model, as well as the comparison of the overall model. The experimental results after removing LSTM (DFNN + STAN) show MAEs of 49.73 and 36.86 on C-MAPSS and PHM08 Challenge Data, respectively, which are significantly higher than the 10.52 and 19.84 of the overall LODS model. This shows that LSTM plays a key role in the model performance when dealing with long-term dependen-

cies in time series data. The experimental results after removing the STAN (DFNN + LSTM) on the Tennessee Eastman Process Simulation Data show an MAPE of 10.89, which is significantly higher than the 5.54 of the LODS model. This shows that the STAN plays a positive role in improving the model performance when considering spatiotemporal relationships. The experimental results after removing the DFNN (STAN + LSTM) show RMSEs of 6.76 and 5.93 on C-MAPSS and PHM08 Challenge Data, respectively, which are higher than the 2.98 and 2.23 of the LODS model. This implies the importance of the DFNN in multi-modal information fusion, especially when processing data such as images. The overall LODS model achieves better performance on all data sets relative to any single-component model. On C-MAPSS and PHM08 Challenge Data, the MAE of LODS is only 10.52 and 19.84, respectively, further verifying its excellent performance in mine maintenance robot tasks.

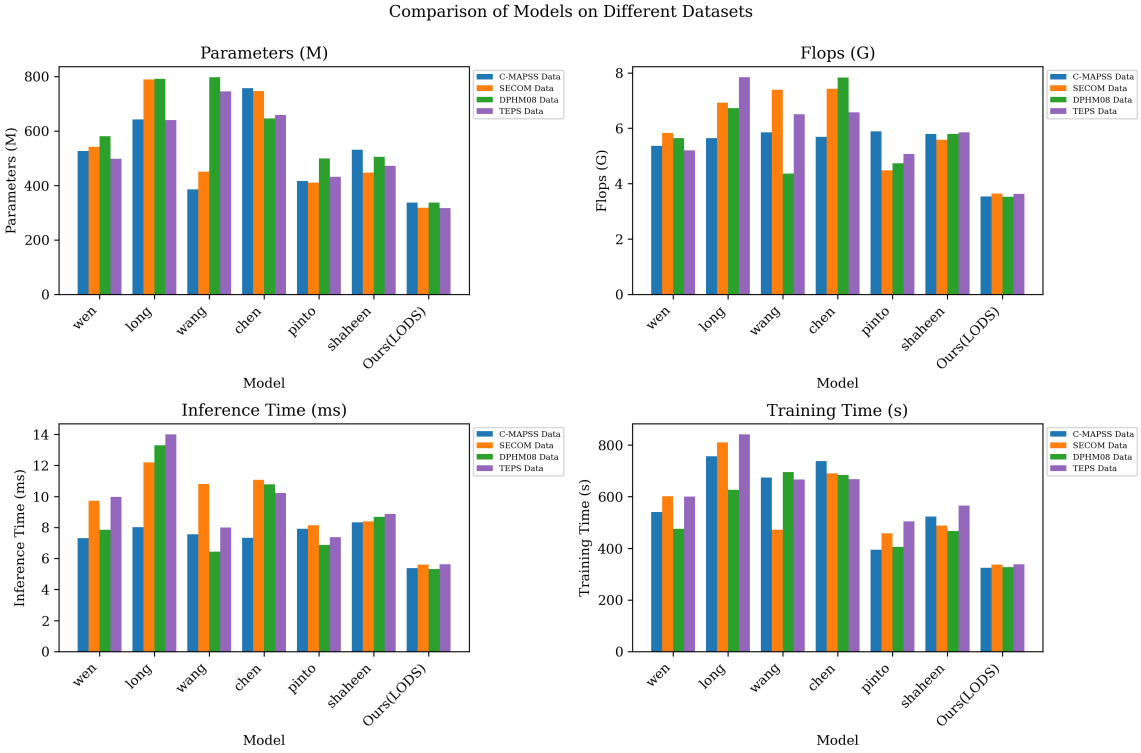

Figure 6. Model efficiency verification comparison chart of different indicators of different models.

Figure 7 shows the comparison of the different models in different performance indicators, which further confirms the significant comprehensive performance advantage of the LODS model compared with each single-component model in the ablation experiment.

As shown in Table 4, we conducted a series of comparative experiments, using Adam, Bayesian and PSO as comparison objects and comparing their performance with that of LSTM. The following conclusions can be drawn from the data in the table. For the Adam vs. LSTM experiment, on C-MAPSS and PHM08 Challenge Data, the LODS model is significantly better than the Adam optimization strategy in indicators such as MAE and RMSE. Especially on C-MAPSS, the MAE of LODS is only 10.52, while that of Adam is 374.35, which shows the excellent performance of LODS in fault prediction tasks. For the Bayesian vs. LSTM experiment, the LODS model achieved better performance on all data sets. Compared with the Bayesian optimization strategy, it performed better in indicators such as MAPE and RMSE. For example, on the PHM08 Challenge Data, the MAPE of LODS is 4.48, while that of the Bayesian is 262.26, further verifying the superiority of LODS. For the PSO vs. LSTM experiment, on C-MAPSS and PHM08 Challenge Data, the LODS model is significantly better than the PSO optimization strategy in indicators such as MAE and RMSE. Especially on C-MAPSS, the MAE of LODS is 10.52, while that of PSO is 312.10,

indicating that LODS is more reliable in fault prediction tasks. Overall, the LODS model has better performance than the optimization strategies in various comparative experiments.

Table 3. Ablation experiments on the LODS model using different data sets.

Data Set	Model	MAE	MAPE	RMSE	MSE
C-MAPSS Data	DFNN + STAN	49.73	10.94	4.64	25.37
	DFNN + LSTM	36.79	9.42	7.75	14.83
	STAN + STAN	26.21	8.72	6.76	14.30
	LODS	10.52	6.36	2.98	11.14
SECOM Data	DFNN + STAN	25.37	30.61	14.30	7.50
	DFNN + LSTM	14.83	37.51	13.15	7.78
	STAN + STAN	14.30	20.41	13.64	7.75
	LODS	18.94	7.49	4.19	6.44
DPHM08 Data	DFNN + STAN	28.63	36.86	15.35	8.30
	DFNN + LSTM	25.63	44.95	9.19	4.86
	STAN + STAN	13.36	50.22	12.01	5.93
	LODS	19.84	4.48	2.23	10.96
TEPS Data	DFNN + STAN	27.92	9.86	6.42	26.98
	DFNN + LSTM	45.01	10.89	7.70	18.33
	STAN + STAN	31.97	11.76	7.90	16.52
	LODS	17.99	5.54	2.62	8.75

Figure 8 visualizes the contents of the table and shows the comparison of the different performance indicators of each model through intuitive charts, further proving the superior performance of the LODS model in mining robot maintenance tasks.

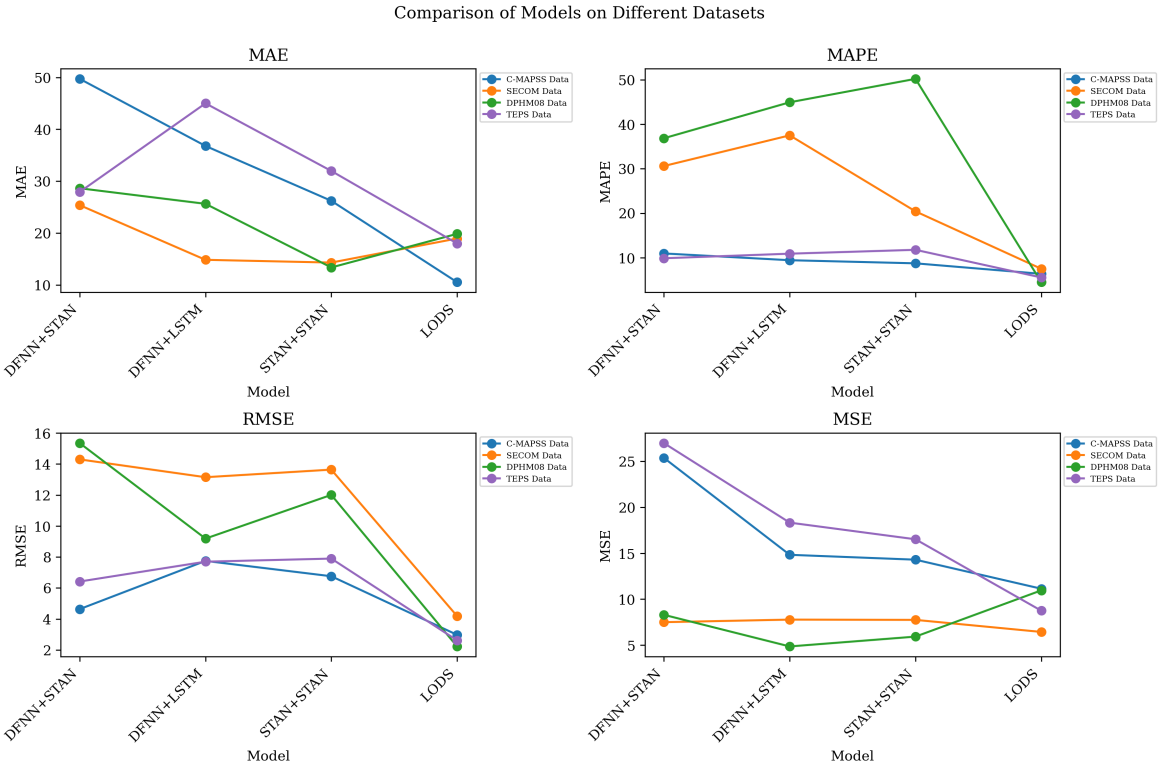

Figure 7. Ablation experiments on the LODS model.

Table 4. Ablation experiments on the LSTM model using different data sets.

Data Set	Model	Parameters (M)	FLOPs (G)	Inference Time (ms)	Training Time (s)
C-MAPSS Data	Adam	374.35	270.63	254.8	305.79
	Bayesian	394.06	300.52	267.24	278.41
	PSO	347.18	365.67	263.08	326.14
	LSTM	204.17	183.11	204.16	228.52
SECOM Data	Adam	375.76	384.71	216.2	405.58
	Bayesian	266.38	359.57	376.29	351.71
	PSO	346.81	321.19	278.58	365.73
	LSTM	163.63	172.28	189.83	125.5
DPHM08 Data	Adam	382.88	306.85	300.45	388.57
	Bayesian	373.02	262.26	252.8	290.94
	PSO	312.1	325.6	233.99	285.37
	LSTM	129.66	133.59	233.97	188.68
TEPS Data	Adam	284.56	257.47	333.25	383.34
	Bayesian	373.29	291.78	224.5	401.1
	PSO	357.89	287.25	395.15	397.16
	LSTM	207.77	220.77	205.69	196.23

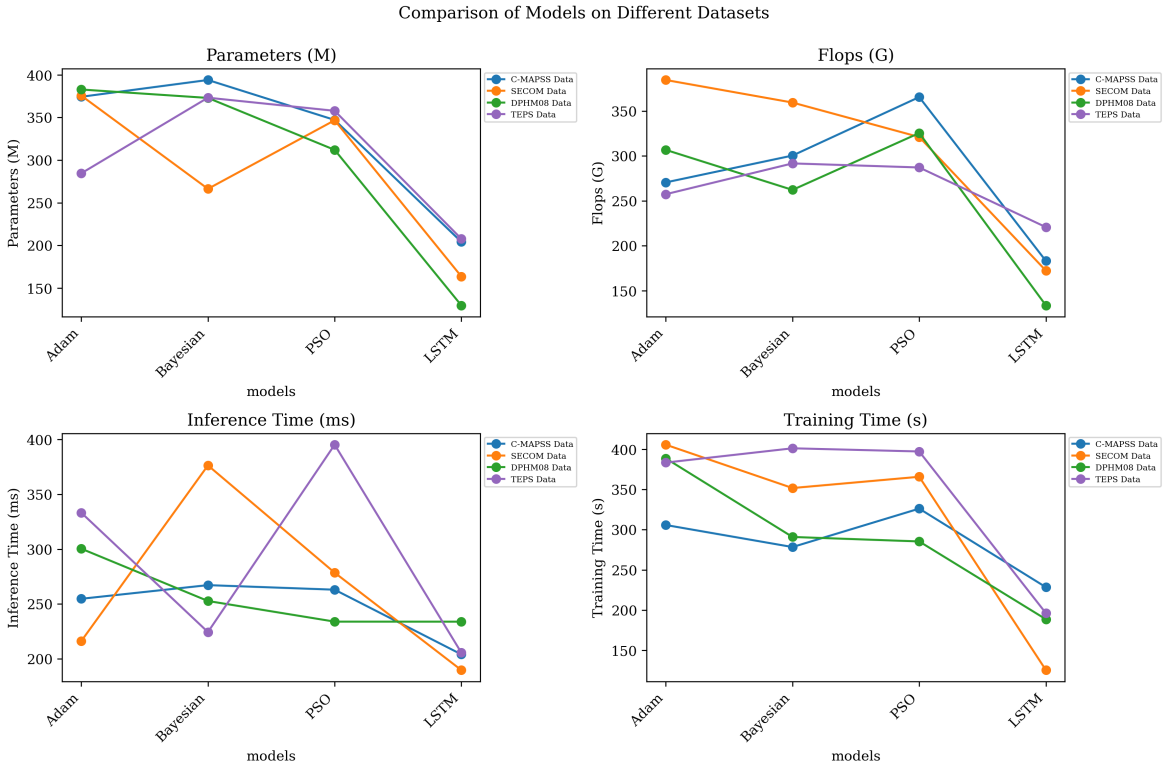

Figure 8. Ablation experiments on the LSTM model.

5. Discussion

In this study, we propose an innovative LODS model designed to improve the accuracy of the predictive modeling of mine robots when performing maintenance tasks through an in-depth analysis of multi-modal temporal data. The model combines the advantages of the long short-term memory network (LSTM), dense feedforward neural network (DFNN) and spatial and temporal attention network (STAN) to specifically solve the long-term dependence problem and the spatiotemporal information processing challenges in mine maintenance. We tested it on multiple industry standard data sets and confirmed that the

LODS model has significant performance improvements compared to existing models and methods in key performance metrics.

The LODS model provides a comprehensive and efficient deep learning solution for the field of mining robot maintenance, and it can more comprehensively understand the equipment status and improve the accuracy and reliability of fault prediction. Its advantages are reflected in the comprehensive analysis of multi-modal time series data, spatiotemporal data modeling and long-term dependency processing, providing more reliable decision support for mine maintenance work.

However, we also recognize that the LODS model may be limited by the quality and diversity of the training data, and the robustness in different mining environments needs to be verified by further studies. In addition, the high computing requirements of the model may pose challenges for real-time deployment on resource-limited robotic platforms.

In order to further develop the LODS model, we plan to conduct research in the following directions: improve the generalization capability and adaptability of the data through advanced data enhancement technology and diverse data integration; optimize the model structure and algorithm, reduce the computational resource demand and support real-time application; deepen the adaptive research on the dynamic mine environment and consider adding an online learning mechanism to timely respond to environmental changes; improve the LODS model to ensure excellent performance in challenging mine maintenance tasks and provide reliable prediction support to maximize the safety and efficiency.

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

Funding: This research was supported by the Science and Technology Innovation Program of Higher Education Institutions (Grant No. 2022L431), the Basic Research Program of Shanxi Province (Grant No. 202303021211325) and the Shanxi Datong University Scientific Research Projects (Grant No. 2022K13).

Data Availability Statement: The datasets generated and/or analyzed during the curent study are not publiclyavailable due to data information that still needs to be organized and improved but areavailable from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Long, J.; Mou, J.; Zhang, L.; Zhang, S.; Li, C. Attitude data-based deep hybrid learning architecture for intelligent fault diagnosis of multi-joint industrial robots. *J. Manuf. Syst.* **2021**, *61*, 736–745. [CrossRef]
2. Saufi, S.R.; Ahmad, Z.A.B.; Leong, M.S.; Lim, M.H. Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: A review. *IEEE Access* **2019**, *7*, 122644–122662. [CrossRef]
3. Fahle, S.; Prinz, C.; Kuhlenkötter, B. Systematic review on machine learning (ML) methods for manufacturing processes–Identifying artificial intelligence (AI) methods for field application. *Procedia CIRP* **2020**, *93*, 413–418. [CrossRef]
4. Ning, X.; Tian, W.; He, F.; Bai, X.; Sun, L.; Li, W. Hyper-sausage coverage function neuron model and learning algorithm for image classification. *Pattern Recognit.* **2023**, *136*, 109216. [CrossRef]
5. Wang, C.; Ning, X.; Sun, L.; Zhang, L.; Li, W.; Bai, X. Learning discriminative features by covering local geometric space for point cloud analysis. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 1–15. [CrossRef]
6. Zhang, C.; Liu, G.; Zhan, X.; Shi, H.; Cai, H.; Li, Y. Multiple Object Tracking Algorithm Based on Mask R-CNN. *J. Jilin Univ. Sci. Ed.* **2021**, *59*, 609–618.
7. Zhi, Z.; Liu, L.; Liu, D.; Hu, C. Fault detection of the harmonic reducer based on CNN-LSTM with a novel denoising algorithm. *IEEE Sens. J.* **2021**, *22*, 2572–2581. [CrossRef]
8. Zhang, P.; Yu, X.; Bai, X.; Wang, C.; Zheng, J.; Ning, X. Joint discriminative representation learning for end-to-end person search. *Pattern Recognit.* **2024**, *147*, 110053. [CrossRef]
9. Xiao, H.; Zeng, H.; Jiang, W.; Zhou, Y.; Tu, X. HMM-TCN-based health assessment and state prediction for robot mechanical axis. *Int. J. Intell. Syst.* **2022**, *37*, 10476–10494. [CrossRef]

10. Liao, Y.; Yeaser, A.; Yang, B.; Tung, J.; Hashemi, E. Unsupervised fault detection and recovery for intelligent robotic rollators. *Robot. Auton. Syst.* **2021**, *146*, 103876. [CrossRef]
11. Ning, X.; Yu, Z.; Li, L.; Li, W.; Tiwari, P. DILF: Differentiable rendering-based multi-view Image–Language Fusion for zero-shot 3D shape understanding. *Inf. Fusion* **2024**, *102*, 102033. [CrossRef]
12. Xia, L.; Zheng, P.; Li, X.; Gao, R.X.; Wang, L. Toward cognitive predictive maintenance: A survey of graph-based approaches. *J. Manuf. Syst.* **2022**, *64*, 107–120. [CrossRef]
13. Zhou, K.; Yang, C.; Liu, J.; Xu, Q. Deep graph feature learning-based diagnosis approach for rotating machinery using multi-sensor data. *J. Intell. Manuf.* **2023**, *34*, 1965–1974. [CrossRef]
14. Ning, E.; Wang, Y.; Wang, C.; Zhang, H.; Ning, X. Enhancement, integration, expansion: Activating representation of detailed features for occluded person re-identification. *Neural Netw.* **2024**, *169*, 532–541. [CrossRef] [PubMed]
15. Chen, C.; Liu, C.; Wang, T.; Zhang, A.; Wu, W.; Cheng, L. Compound fault diagnosis for industrial robots based on dual-transformer networks. *J. Manuf. Syst.* **2023**, *66*, 163–178. [CrossRef]
16. Singh, A.; Patil, A.J.; Jarial, R. A Fuzzy Modeling Technique to Assist Submersible Inspection Robot for Internal Inspection of Transformers. In Proceedings of the 2020 Fourth International Conference on Inventive Systems and Control (ICISC), IEEE, Coimbatore, India, 8–10 January 2020; pp. 284–290.
17. Yu, Z.; Tiwari, P.; Hou, L.; Li, L.; Li, W.; Jiang, L.; Ning, X. MV-ReID: 3D Multi-view Transformation Network for Occluded Person Re-Identification. *Knowl.-Based Syst.* **2024**, *283*, 111200. [CrossRef]
18. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
19. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
20. Chen, J.; Ma, T.; Xiao, C. FastGCN: Fast learning with graph convolutional networks via importance sampling. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
21. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
22. Saha, A.; Saha, J.; Mallik, M.; Chowdhury, C. AI Enabled Human and Machine Activity Monitoring in Industrial IoT Systems. In *AI Models for Blockchain-Based Intelligent Networks in IoT Systems: Concepts, Methodologies, Tools, and Applications*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 29–54.
23. Lee, C.S.; Tsai, Y.L.; Wang, M.H.; Kuan, W.K.; Ciou, Z.H.; Kubota, N. AI-FML agent for robotic game of Go and AIoT real-world co-learning applications. In Proceedings of the 2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), IEEE, Glasgow, UK, 19–24 July 2020; pp. 1–8.
24. He, Q.; Pang, Y.; Jiang, G.; Xie, P. A spatio-temporal multiscale neural network approach for wind turbine fault diagnosis with imbalanced SCADA data. *IEEE Trans. Ind. Inform.* **2020**, *17*, 6875–6884. [CrossRef]
25. Ding, C.; Sun, S.; Zhao, J. MST-GAT: A multimodal spatial–temporal graph attention network for time series anomaly detection. *Inf. Fusion* **2023**, *89*, 527–536. [CrossRef]
26. Han, P.; Ellefsen, A.L.; Li, G.; Holmeset, F.T.; Zhang, H. Fault detection with LSTM-based variational autoencoder for maritime components. *IEEE Sens. J.* **2021**, *21*, 21903–21912. [CrossRef]
27. Zheng, Y.; Zuo, X.; Zuo, W.; Liang, S.; Wang, Y. Bi-LSTM+GCN Causality Extraction Based on Time Relationship. *J. Jilin Univ. Sci. Ed.* **2021**, *59*, 643–648.
28. Li, L.; Liu, J.; Wei, S.; Chen, G.; Blasch, E.; Pham, K. Smart robot-enabled remaining useful life prediction and maintenance optimization for complex structures using artificial intelligence and machine learning. In Proceedings of the Sensors and Systems for Space Applications XIV, SPIE, Online, 26 April 2021; Volume 11755, pp. 100–108.
29. Çınar, Z.M.; Abdussalam Nuhu, A.; Zeeshan, Q.; Korhan, O.; Asmael, M.; Safaei, B. Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability* **2020**, *12*, 8211. [CrossRef]
30. ElDali, M.; Kumar, K.D. Fault diagnosis and prognosis of aerospace systems using growing recurrent neural networks and lstm. In Proceedings of the 2021 IEEE Aerospace Conference (50100), IEEE, Big Sky, MT, USA, 6–13 March 2021; pp. 1–20.
31. Lomov, I.; Lyubimov, M.; Makarov, I.; Zhukov, L.E. Fault detection in Tennessee Eastman process with temporal deep learning models. *J. Ind. Inf. Integr.* **2021**, *23*, 100216. [CrossRef]
32. Wen, Y.; Tang, Z.; Pang, Y.; Ding, B.; Liu, M. Interactive spatiotemporal token attention network for skeleton-based general interactive action recognition. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, Detroit, MI, USA, 1–5 October 2023; pp. 7886–7892.
33. Long, J.; Qin, Y.; Yang, Z.; Huang, Y.; Li, C. Discriminative feature learning using a multiscale convolutional capsule network from attitude data for fault diagnosis of industrial robots. *Mech. Syst. Signal Process.* **2023**, *182*, 109569. [CrossRef]
34. Wang, X.; Liu, M.; Liu, C.; Ling, L.; Zhang, X. Data-driven and Knowledge-based predictive maintenance method for industrial robots for the production stability of intelligent manufacturing. *Expert Syst. Appl.* **2023**, *234*, 121136. [CrossRef]
35. Chen, S.; Cheng, L.; Deng, J.; Wang, T. Multi-Feature Fusion Event Argument Entity Recognition Method for Industrial Robot Fault Diagnosis. *Appl. Sci.* **2022**, *12*, 12359. [CrossRef]

36. Pinto, R.; Cerquitelli, T. Robot fault detection and remaining life estimation for predictive maintenance. *Procedia Comput. Sci.* **2019**, *151*, 709–716. [CrossRef]
37. Shaheen, K.; Hanif, M.A.; Hasan, O.; Shafique, M. Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks. *J. Intell. Robot. Syst.* **2022**, *105*, 9. [CrossRef]

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.