Electrical-STGCN: An Electrical Spatio-Temporal Graph Convolutional Network for Intelligent Predictive Maintenance

***Abstract*— With the rapid improvement of industrial Internet of Things and artificial intelligence, predictive maintenance (PdM) has attracted concerns from both academia and industry. While equipment is working, the electrical attributes have intrinsic relations. Meanwhile, they are changing over the time. However, existing PdM models are often limited by a lack consideration of both attribute interactions and temporal dependency of the dynamic working system. To address the problem, this paper proposes an Electrical Spatio-Temporal Graph Convolutional Network (Electrical-STGCN) for health estimation. It takes a sequence of electrical records as input. Next, both attribute interactions and temporal dependency are established to extract features. Then, the extracted features are fed into a health prediction component. Finally, the output of the Electrical-STGCN (i.e., Remaining Useful Life) can help the workers decide whether to carry out equipment maintenance. The effectiveness of the proposed method is verified with real-world cases. Our method achieves 85.2% Accuracy and 0.9 F1-Score, which is the current state-of-the-art performances.**

***Index Terms*— industrial Internet of Things, artificial intelligence, predictive maintenance, temporal dependency, attribute interactions**

I. Introduction

Predictive maintenance (PdM), a data-driven health prediction strategy, is widely employed in various of fields such as public transport [1], pharmaceutical systems [2], and semiconductor manufacturing [3] on grounds of minimizing operation costs, making reliable predictions, and improving equipment conditions. Firstly, it can minimize operating costs by using machine learning algorithms on enormous data streams to determine when maintenance activities are necessary. Secondly, it can make more reliable predictions by considering historical data, engineering approaches, and statistical methods. Thirdly, it can improve working conditions of the industrial equipment because it can deal with faults detection before they happen. Thus, research shows that PdM is the current state-of-the-art choice for maintenance of industrial equipment.

In the context of Industry 4.0, physical and digital systems are converging to provide novel PdM solutions. It is related to a systemic adoption of Internet-of-Things (IoT), 5G/6G technology, and artificial intelligence [4]. Ayvaz and Alpay [5] develop an IoT-based PdM system in a manufacturing system. By utilizing data collected from the IoT sensors, the proposed system can predict potential failures with machine learning algorithms. Cheng et al. [6] propose a data-driven PdM planning framework for civil engineering based on IoT technology and machine learning. The framework contains an information layer and an application layer. Data collection and integration are conducted in the information layer while the application layer is applied for PdM. To improve data transmission efficiency, keep data significance, and reduce storage amounts in the wireless sensor networks, Chen et al. [7] provide a spatiotemporal data compression approach with low transmission cost and high data fidelity. Information security is another crucial part in the IoT networks. Lin et al. [8] propose a privacy-preserving multiobjective model in 6G IoT environments. The latest research achievements on IoT and data communication provide more solutions for PdM.

Existing PdM approaches can be grouped into statistical approaches and artificial intelligence (AI) based approaches [9]. The statistical approaches establish explicit mathematical models for operational capability degradation. Depending on how to obtain the condition monitoring data, the statistical approaches can be classified into Direct approaches and Indirect approaches [10]. The Direct approaches include Bayesian models [11], regression-based models [12], Gamma processed [13], and Markov-based models [14]. The Indirect approaches include stochastic filtering-based models [15], covariated based hazard models [16], and Hidden Markov Models (HMM) [17]. Implementing statistical approaches requires five generic steps, i.e., preliminary data analysis, health indicators construction, degradation model establishment, health prediction, and evaluation [18]. Although statistical models perform well, it is difficult to establish detailed mathematical model for complex working process. Additionally, statistical models are complicated because they require professional background such as mathematical and mechanical knowledge. Moreover, the over-reliance on expertise is time-consuming and unsuitable for huge amounts of sensor data streams.

The AI-based approaches monitor the manufacturing systems with smart sensors in real-time and estimate the health condition with machine learning (ML) algorithms to provide intelligent PdM service. With the improvement of industrial Internet of Things (IIoT) and artificial intelligence, the AI-based approaches outperform the statistical approaches in terms of dealing with real-time environments, handling high-dimensional data, and capturing valuable insights. Although the AI-based PdM is the most effective for health prediction, there exist some challenges that need to be overcome. Take current and voltage of the working equipment for an example. These electrical attributes have intrinsic relations, meanwhile, they are changing over time. Thus, it is necessary to consider

both attribute interactions and temporal dependency of the monitored system for health estimation.

However, existing AI-based approaches lack consideration of attribute interactions of industrial equipment, which affects the performance of PdM model. While the equipment is working, there are potential interactions between each attribute. ML algorithms employed in PdM focus more on extracting features from the monitoring data whereas neglect the transfer relationships between the physical attributes.

Other than that, most AI-based approaches lack the consideration of temporal dependency of the manufacturing system, which makes the detection less reliable. The health condition depends on both real-time and historical data because current health condition is affected by the previous. Hence, it cannot always determine the health condition by a single record but needs the inter-correlations of a relatively long sequence.

To solve these problems, we propose an Electrical Spatio-Temporal Graph Convolutional Network (Electrical-STGCN) for Remaining Useful Life (RUL) estimation. Three steps are required to implement the model. Firstly, electrical data is consistently collected as input. Secondly, attribute interactions and temporal dependency of the working system are established by a spacio-temporal graph convolutional network to extract features. Thirdly, the features are fed into a health prediction network for RUL estimation. The contributions of this paper are listed as follows:

1) We propose an Electrical-STGCN for intelligent PdM of the manufacturing system. Different from the existing works, both attribute interactions and temporal dependency are considered for current health estimation.
2) We introduce a novel kernel function to address over-smoothing in the graph-based PdM application. It is calculated by the eigenvector which has the lowest non-zero eigenvalue of the symmetric-normalized Laplacian matrix. We also provide a similarity kernel function to model the attribute interactions.
3) The effectiveness of Electrical-STGCN is verified with real-world cases. The results suggest that our approach achieves the current state-of-the-art performance.

The rest of this paper is organized as follows. Section II gives a literature review about commonly used ML algorithms in PdM works. Section III describes the proposed method. Experiments and analysis are presented in Section IV. Finally, conclusion and future work are discussed in Section V.

II. Literature Review

The Remaining Useful Life (RUL) refers to the remaining service life of the system after a period of operation. Accurate RUL estimation is of crucial importance in the manufacturing system where high reliability is required. With the improvement of data analytics, various of ML algorithms are employed in PdM works to improve the prediction results. These algorithms can be grouped into four main categories, i.e., Supporting Vector Machines (SVM), K-Means, Ensemble Learning (EL), and Artificial Neural Network (ANN).

SVM is a well-known supervised algorithm for classification and regression tasks because of its high precision [19]. Susto et al. [20] exploit an SVM model to separate faulty and non-faulty runs of the equipment in semiconductor manufacturing. In this study, 3671 records are collected and analyzed. The proposed model is trained through Monte Carlo cross-validation (MCCV) and tested on a real production dataset. Li et al. [21] propose an SVM model for faults detection in the rail network to increase network velocity, avoid interruptions and increase safety. In this work, huge volumes of historical data, failure data, weather data and maintenance data are used to predict truck performance alarms leading to failure. Despite the promising results of SVM, there are some disadvantages. Firstly, it lacks temporal dependency while making decision on performing the maintenance activity or not. Secondly, it is difficult to be optimized in a complex condition where the cost of unexpected breaks is difficult to estimate.

K-Means is a popular unsupervised algorithm for clustering tasks on grounds of good performance and easy-to-understand [22]. It aims to find K clusters in which samples are "close" to each other, so "far" samples can be regarded as anomalies. Uhlmann et al. [23] propose a K-Means algorithm for faults detection in a selective laser melting machine tool. In this work, platform temperature, oxygen percentage, and process chamber pressure are manually selected. The proposed model can identify four target conditions. Although K-Means is easy to implement, it has some drawbacks. Firstly, it is difficult to determine the number of clusters. Secondly, it is order sensitive, i.e., the data entry order will cause changes to the final results. Thirdly, different data normalizing strategy will also impact the final results.

EL algorithms generate many weak classifiers and aggregate them together to form better results [24]. Two famous algorithms of EL are Random Forests (RFs) and AdaBoost. Canizo et al. [25] present a cloud-deployed Big Data architecture using RFs algorithm to predict failures on wind turbines. In this work, status and operational data are gathered and only status data is selected. It has already been demonstrated that one company's productivity has increased 3% after employing this framework. Li et al. [26] propose a novel non-convex archetypal one-class classification algorithm, which combines the random projection and the AdaBoost algorithm for anomaly detection in cyber-physical systems. In this study, random projections are used to detect anomalies in arbitrary dimension and the AdaBoost is used to adaptively select the appropriate directions of the projections. The proposed method is evaluated with both artificial and real world datasets and the result is promising. To the best of our knowledge, EL is one of the most used and compared ML algorithm in PdM. However, it has some shortcomings, for instance, it is complex in practice and time-consuming in computation.

ANN, inspired by the biological neurons, has been widely used in PdM works on grounds of handling high-dimensional data, extracting valuable features, and dealing with real-time environments. Olgun and Seren [27] provide a Long Short-Term Memory (LSTM) network, a type of Recurrent Neural Network (RNN), on the Apache Spark framework for engine anomaly prediction. In this work, 21 physical attributes are gathered and 10 of them are manually selected as the input. The accuracy indicates the reliability of the proposed model.

Another variant of the RNN, known as Gated Recurrent Units (GRU), is quite simpler to implement than LSTM as it has less number of gates. Lu et al. [28] propose an Autoencoder Gated Recurrent Units (AE-GRU) model for RUL prediction. In this work, important features are extracted by the autoencoder, and then fed into the GRU. The AE-GRU has good accuracy in predicting RUL. Essien and Giannetti [29] propose a deep convolutional LSTM (ConvLSTM) architecture for sequential forecasting problems in the smart factories. In this work, the internal speed data of a metal can bodymaker machine is utilized as input and the sliding-window method is employed. The effectiveness of ConvLSTM is demonstrated by real-world data collected from a metal packaging plant. Wang et al. [30] introduce a Temporal Convolutional Network (TCN) for machinery prognostics. In this work, condition monitoring data are directly used for RUL estimation. Zhang et al. [31] and Wang et al. [32] propose the spatio-temporal graph neural network (STGCNN) to estimate the RUL of aircraft engines. They construct graph convolutional network on spatial learning and temporal convolutional network on sequence learning. The experimental results demonstrate that the established spatio-temporal graph structure can model the system accurately as well as improve the performance of RUL estimation. However, the graph-based networks have the problem of over-smoothing.

To sum up, one common disadvantage of SVM, K-Means, and EL is lacking temporal dependency of the working system. Because current health condition is affected by the previous, determining health condition by a single record is less reliable. The RNNs focus more on extracting temporal features whereas neglect the potential transfer relationship between different attributes, leading to a loss of attribute dependency information. Graph-based methods face the problem of over-smoothing. To address these issues, we employ an Electrical-STGCN, which considers both attribute interactions and temporal dependency of the working system for RUL estimation. Besides, we design a novel kernel function to address the over-smoothing problem.

III. The Proposed Electrical-STGCN

A. Data acquisition

To support the PdM applications, we collected the data from a horizontal machining center in a manufacturing corporation. The three-phase circuit wiring diagram is shown in Fig. 1.

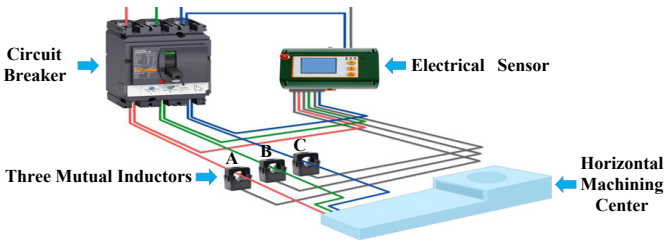

Fig. 1. **Equipment layout.**

Raw electrical data is collected by the smart sensor attached to the industrial equipment. After data pre-processing, current health condition is estimated by the proposed algorithm to support the workers to carry out equipment maintenance. Within the practical three-phase circuit, it is dangerous to measure the high-current and the high-voltage directly. Thus, three mutual inductors that responsible for A phase, B phase, and C phase are utilized to transform high-current and high-voltage to low-current and low-voltage respectively. Additionally, the function of the circuit breaker is to prevent current and voltage overload so as to ensure the safety of the workers and devices.

The dataset consists of ten maintenance cycles using run-to-fail (R2F) policy while processing different work-pieces and the sampling rate is 50Hz. In the first eight maintenance cycles, anomalies occurred during the working process. In the last two maintenance cycles, random anomalies were captured when the equipment was in off-working condition. The details of each maintenance cycle are presented in Table I.

TABLE I
DETAILS OF EACH MAINTENANCE CYCLE (MC).

MC	1	2	3	4	5	6	7	8	9	10
k^{th}	1558	126	1714	1747	4415	2810	1063	3756	6289	10916

$No.k^{th}$ is an anomaly observed in each MC. There is a total of 34394 records and each record contains 15 attributes. Data range, mean value, and standard deviation of each signal (i.e., attribute) are shown in Table II. Health condition (i.e., RUL) is calculated according to these records. When the RUL reaches a predefined alarm threshold ξ, it means that the system needs maintenance [33].

TABLE II
SPECIFICATIONS OF THE TOTAL 34394 RECORDS.

Attribute	Variable	Range	Mean/Standard Deviation
A phase power factor	$\cos\varphi_A$	0.40-1	0.84/0.17
B phase power factor	$\cos\varphi_B$	0.27-1	0.82/0.20
C phase power factor	$\cos\varphi_C$	0.31-1	0.78/0.23
A phase reactive power	Q_A	1-133 W	25.21/25.11
B phase reactive power	Q_B	3-152 W	32.76/32.05
C phase reactive power	Q_C	1-137 W	30.79/33.02
A phase current	I_A	0-34.8 A	6.58/7.05
B phase current	I_B	0.4-40 A	8.03/8.69
C phase current	I_C	0-38 A	7.11/7.83
A phase voltage	U_A	213.2-229.9 V	223.20/3.03
B phase voltage	U_B	217.6-231.7 V	226.74/2.39
C phase voltage	U_C	217.9-231.4 V	226.34/2.53
A phase frequency	f_A	49.96-50.03 Hz	49.99/0.02
B phase frequency	f_B	25.02-50.03 Hz	49.99/0.14
C phase frequency	f_C	49.96-50.03 Hz	49.99/0.02

B. Problem formulation and data transformation

Given an electrical data sequence $\mathbf{X}_\tau = \{\mathbf{X}^{t-\tau+1}, ..., \mathbf{X}^t\} \in \mathbb{R}^{\tau\times 15}$ of τ past time periods (i.e., window size τ), where $\mathbf{X}^i = (\cos\varphi_A{}^i, \cos\varphi_B{}^i, \cos\varphi_C{}^i, Q_A^i, Q_B^i, Q_C^i, I_A^i, I_B^i, I_C^i, U_A^i, U_B^i, U_C^i, f_A^i, f_B^i, f_C^i) \in \mathbb{R}^{1\times 15}$, $i \in \{t-\tau+1, ..., t\}$, our target is to estimate the current health condition $\widehat{RUL}^t$.

To establish the potential interactions between attributes and their temporal dependency in the dynamic working system, $\mathbf{X}_\tau$ should be organized into the spatio-temporal graph G_τ, which is a stack of spatial graphs $\{G^{t-\tau+1}, ..., G^t\}$ corresponding to $\{\mathbf{X}^{t-\tau+1}, ..., \mathbf{X}^t\}$ at each time step. We take the $\mathbf{X}^t \rightarrow G^t$ for an example to explain how to organize $\mathbf{X}_\tau$ into G_τ.

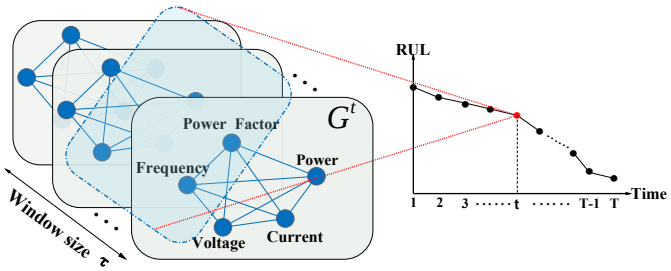

Fig. 2. **Spatio-temporal graph representation of the X_τ for RUL estimation.**

As shown in Fig. 2, G^t is defined as:

$$G^t = (V^t, E^t, \mathbf{A}^t), \tag{1}$$

where V^t=$\{v^t_{power\ factor}$,v^t_{power},$v^t_{current}$,$v^t_{voltage}$,$v^t_{frequency}\}$ is a set of vertices. E^t is the set of edges expressed as $\mathbf{E}^t = \{e^t_{jk} | \forall j,k \in V^t\} \in \mathbb{R}^{5\times5}$. $e^t_{jk} = 1$ if j,k are connected, $e^t_{jk} = 0$ otherwise. $\mathbf{A}^t \in \mathbb{R}^{5\times5}$ is the adjacency matrix.

We assign $\cos\varphi_A{}^t$, $\cos\varphi_B{}^t$, and $\cos\varphi_C{}^t$ to $v^t_{power\ factor}$; Q^t_A, Q^t_B, and Q^t_C to v^t_{power}; I^t_A, I^t_B, and I^t_C to $v^t_{current}$; U^t_A, U^t_B, and U^t_C to $v^t_{voltage}$; f^t_A, f^t_B, and f^t_C to $v^t_{frequency}$. Thus, the matrix representation of V^t in each G^t is formulated as:

$$\mathbf{V}^t = \begin{pmatrix} \cos\varphi^t_A & \cos\varphi^t_B & \cos\varphi^t_C \\ Q^t_A & Q^t_B & Q^t_C \\ I^t_A & I^t_B & I^t_C \\ U^t_A & U^t_B & U^t_C \\ f^t_A & f^t_B & f^t_C \end{pmatrix} \in \mathbb{R}^{5\times3}. \tag{2}$$

It is worth noticing that the topology of G^t keeps the same while the attribute values are different when t varies. In addition, We let G^t be full-connected (i.e. $\mathbf{E}^t = \mathbf{1}$) so as to model the potential interactions between different vertices. In order to model how strongly two vertices interact with each other, we attach a value a^t_{jk}, which is computed by some kernel functions for each e^t_{jk}. The a^t_{jk}s are organized into the weighted adjacency matrix $\mathbf{A}^t$.

C. Kernel functions

In this section, we design the $a^t_{sim(j,k)}$ and $a^t_{\phi_1(j,k)}$ where $\forall j,k \in V^t$ as kernel functions to compute the weighted adjacency matrices $\mathbf{A}^t_{sim}$ and $\mathbf{A}^t_{\phi_1}$. The $\mathbf{A}^t_{sim}$ describes the similarity between different vertices (i.e., $v^t_{power\ factor}$, v^t_{power}, $v^t_{current}$, $v^t_{voltage}$, and $v^t_{frequency}$) at time t to model potential interactions. The $\mathbf{A}^t_{\phi_1}$ is calculated by the eigenvector which has the lowest non-zero eigenvalue of the symmetric-normalized Laplacian matrix to reduce over-smoothing. We combine them for message passing in each spatial graph G^t.

A straightforward idea in designing the kernel function is to use the distance measured by L_2 norm. However, it is against the intuition that vertices tend to be influenced more by the closer ones. To solve this problem, we use the inverse of L_2 norm to measure the similarity between vertices defined as:

$$a^t_{sim(j,k)} = \begin{cases} 1/||\boldsymbol{v}^t_j - \boldsymbol{v}^t_k||_2, & ||\boldsymbol{v}^t_j - \boldsymbol{v}^t_k||_2 \neq 0, \\ 0, & otherwise. \end{cases} \tag{3}$$

The other problem is that in most graph neural networks (GNNs), vertex representations become over-smoothed after several rounds of message passing (i.e. convolutions) as the representations tend to be reach a mean equilibrium equivalent to the stationary distribution of a random walk. To solve this problem, we use the eigenvector ϕ^t_1, which has the lowest non-zero eigenvalue of the symmetric-normalized Laplacian matrix shown in Fig. 3.

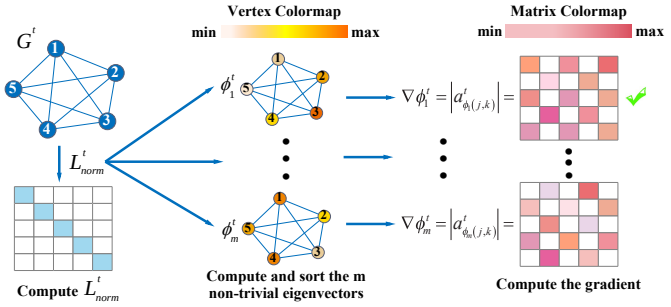

Fig. 3. **Overview of the steps required to compute $a^t_{\phi(j,k)}$.** Number 1 to 5 correspond to vertices power factor, power, current, voltage, and frequency respectively.

First, we compute the symmetric-normalized Laplacian matrix $\mathbf{L}^t_{norm}$ using $\mathbf{E}^t$ and degree matrix $\mathbf{D}^t$ formulated as:

$$\mathbf{L}^t_{norm} = (\mathbf{D}^t)^{-\frac{1}{2}}(\mathbf{D}^t - \mathbf{E}^t)(\mathbf{D}^t)^{-\frac{1}{2}} \in \mathbb{R}^{5\times5}, \tag{4}$$

where both $\mathbf{E}^t$ and $\mathbf{D}^t$ are of size 5×5. Then, the eigenvectors $\boldsymbol{\phi}^t$ of the $\mathbf{L}^t_{norm}$ are computed and sorted so that $\boldsymbol{\phi}^t_1 \in \mathbb{R}^{1\times5}$ has the lowest non-zero eigenvalue and $\boldsymbol{\phi}^t_m \in \mathbb{R}^{1\times5}$ has the m-th lowest. After that, we select $\boldsymbol{\phi}^t_1$ and compute its gradient (a matrix) formulated as:

$$a^t_{\boldsymbol{\phi}_1(j,k)} = \begin{cases} \boldsymbol{\phi}^t_{1(j)} - \boldsymbol{\phi}^t_{1(k)}, & e_{jk} = 1, \\ 0, & otherwise. \end{cases} \tag{5}$$

The visual interpretation of $a^t_{sim(j,k)}$ and $a^t_{\boldsymbol{\phi}_1(j,k)}$ are shown in Fig. 4.

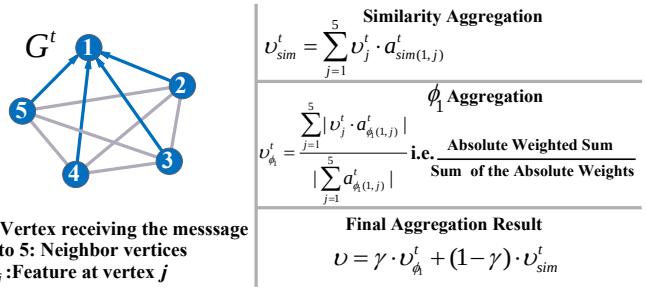

Fig. 4. **Illustration of how the proposed kernel functions work at one vertex.** The final aggregation result is the weighted average of similarity aggregation and ϕ_1 aggregation. γ is a hyper parameter.

We also give theoretical support for the choice of $a^t_{\phi_1(j,k)}$ can address over-smoothing while using the K-walk distance to measure the difficulty of message passing. Thus, we provide a general way to solve the over-smoothing in graph-based PdM.

Definition 1 (K-walk distance). *Suppose j and k are two vertices distributed on a graph. K-walk distance $d_K(j,k)$ is the number of times that k is hit while starting from j and randomly moving K steps.*

Definition 2 (Hitting time). *Suppose starting from vertex j and ending to vertex k on a graph. Hitting time $Q(j,k)$ is the expected number of steps in a random walk. The transition probability of each vertex is $P=\frac{1}{d}$, where d is node degree.*

Conjecture 1 ($a^t_{\phi_1(j,k)}$ can reduce expected hitting time). *Suppose that j, k are uniformly distributed random vertices such that $\phi^t_{1(j)} < \phi^t_{1(k)}$. Let h be the vertex obtained from j by taking one step, then the expected hitting time is decreased proportionally to $(\lambda^t_1)^{-1}$ and $\mathbb{E}_{j,k}[Q(h,k)] \leq \mathbb{E}_{j,k}[Q(j,k)]$.*

Proof: In [34], the hitting time $Q(j,k)$ is given by:

$$Q(j,k) = vol \cdot (\frac{\mathbf{G}(k,k)}{d_k} - \frac{\mathbf{G}(j,k)}{d_j}), \tag{6}$$

with λ^t_m and $\boldsymbol{\phi}^t_m$ being the m-th eigenvalue and eigenvector of the symmetric-normalized Laplacian matrix $\mathbf{L}^t_{norm}$, vol the sum of the degrees of all vertices, and $\mathbf{G}$ the Green's function for the graph formulated as:

$$\mathbf{G}(j,k) = d_j^{\frac{1}{2}} d_k^{-\frac{1}{2}} \sum_{m>0} \frac{1}{\lambda^t_m} \boldsymbol{\phi}^t_{m(j)} \boldsymbol{\phi}^t_{m(k)}. \tag{7}$$

We use $\boldsymbol{\phi}^t_{m(j)} < \boldsymbol{\phi}^t_{m(k)}$ to simplify the argument without having to consider the change in sign since the direction of the eigenvector is not deterministic. Supposing $\lambda^t_1 \ll \lambda^t_2$, the first term of equation (7) has much more weight than the following terms. Since h is the vertex obtained from j by taking one step in the direction of the gradient of $\boldsymbol{\phi}^t_1$, we have:

$$\boldsymbol{\phi}^t_{1(h)} - \boldsymbol{\phi}^t_{1(j)} > 0. \tag{8}$$

We want to show that the following inequality holds:

$$\mathbb{E}_{j,k}[Q(h,k)] < \mathbb{E}_{j,k}[Q(j,k)]. \tag{9}$$

It is equivalent to:

$$\mathbb{E}_{j,k}[\mathbf{G}(h,k)] > \mathbb{E}_{j,k}[\mathbf{G}(j,k)]. \tag{10}$$

By the hypothesis $\lambda^t_1 \ll \lambda^t_2$, we can approximate $\mathbf{G}(j,k) \sim d_j^{\frac{1}{2}} d_k^{-\frac{1}{2}} \frac{1}{\lambda^t_1} \boldsymbol{\phi}^t_{1(j)} \boldsymbol{\phi}^t_{1(k)}$, so inequality (10) is equivalent to:

$$\mathbb{E}_{j,k}[d_h^{\frac{1}{2}} d_k^{-\frac{1}{2}} \frac{1}{\lambda^t_1} \boldsymbol{\phi}^t_{1(h)} \boldsymbol{\phi}^t_{1(k)}] > \mathbb{E}_{j,k}[d_j^{\frac{1}{2}} d_k^{-\frac{1}{2}} \frac{1}{\lambda^t_1} \boldsymbol{\phi}^t_{1(j)} \boldsymbol{\phi}^t_{1(k)}] \tag{11}$$

Removing all equal terms of inequality (11) from both sides, it is equivalent to:

$$\mathbb{E}_{j,k}[d_h^{\frac{1}{2}} \boldsymbol{\phi}^t_{1(h)}] > \mathbb{E}_{j,k}[d_j^{\frac{1}{2}} \boldsymbol{\phi}^t_{1(j)}]. \tag{12}$$

From inequality (8), we know that $\boldsymbol{\phi}^t_{1(h)} > \boldsymbol{\phi}^t_{1(j)}$, and from the choice of h being a step in the direction of $\nabla \boldsymbol{\phi}^t_1$, we believe that $\mathbb{E}(d_h) > \mathbb{E}(d_j)$. Thus, we also believe that **conjecture 1** should hold in general.

In conclusion, using the $\mathbf{A}^t_{\boldsymbol{\phi}_1}$ reduces the influence between different vertex from the random-walk hitting times so as to reduce over-smoothing.

D. Electrical-STGCN realization

As shown in Fig. 5, the Electrical-STGCN consists of two main components, i.e., the Spatio-temporal Graph Convolution Network (STGCN) and the Health Prediction Network (HPN). The STGCN conducts spatio-temporal convolution operations on the graph representation of the electrical sequence to extract feature $\overline{\mathbf{V}_\tau}$, which involves attribute interactions and temporal dependency of the observed records. The HPN takes the $\overline{\mathbf{V}_\tau}$ as input to estimate the RUL of the working system.

Overall, there are two main differences between the proposed Electrical-STGCN and traditional Spatio-Temporal Convolution Neural Network [32]. Firstly, our proposed Electrical-STGCN constructs the graph representation of electrical data sequence in a totally different way with $a^t_{sim(j,k)}$ and $a^t_{\boldsymbol{\phi}_1(j,k)}$ kernel functions. Secondly, beyond STGCN, the HPN is added for health estimation. There are three steps that required to implement the Electrical-STGCN.

1) **Data normalization**. The first step is to normalize $\mathbf{A}_{sim}$, $\mathbf{A}_{\boldsymbol{\phi}_1} \in \mathbb{R}^{\tau\times5\times5}$ computed by equation (3) and equation (5). The $\mathbf{A}_{sim}$ is a stack of $\{\mathbf{A}^{t-\tau+1}_{sim}, ..., \mathbf{A}^t_{sim}\}$. We symmetrically normalize each $\mathbf{A}^t_{sim} \in \mathbb{R}^{5\times5}$ formulated as:

$$\mathbf{A}^t_{sim} = (\mathbf{\Lambda}^t_{sim})^{-\frac{1}{2}} \mathbf{A}^t_{sim} (\mathbf{\Lambda}^t_{sim})^{-\frac{1}{2}}, \tag{13}$$

where $\mathbf{\Lambda}^t_{sim}$ is the diagonal vertex degree matrix of the $\mathbf{A}^t_{sim}$. $\mathbf{A}_{\boldsymbol{\phi}_1}$ is a stack of $\{\mathbf{A}^{t-\tau+1}_{\boldsymbol{\phi}_1}, ..., \mathbf{A}^t_{\boldsymbol{\phi}_1}\}$. For each $\mathbf{A}^t_{\boldsymbol{\phi}_1} \in \mathbb{R}^{5\times5}$, we normalize each row by its L_1 norm to assign larger weight to the elements in the forward or backward message passing direction with a total weight of 1, which is formulated as:

$$\mathbf{A}^t_{\boldsymbol{\phi}_1} = \frac{|\mathbf{A}^t_{\boldsymbol{\phi}_1}(n,:)|}{\left\|\mathbf{A}^t_{\boldsymbol{\phi}_1}(n,:)\right\| + \epsilon}, \tag{14}$$

where n is row number, and ϵ is an arbitrarily small positive number used to avoid floating-point errors. In addition, we normalize each $\mathbf{V}^t \in \mathbb{R}^{5\times3}$ and then stack them orderly into $\mathbf{V}_\tau \in \mathbb{R}^{\tau\times5\times3}$.

2) **Spatio-temporal graph convolution**. The second step is to conduct the spatio-temporal graph convolution. It takes $\mathbf{V}_\tau$, $\mathbf{A}_{sim}$, and $\mathbf{A}_{\boldsymbol{\phi}_1}$ as input. The output is feature $\overline{\mathbf{V}_\tau} \in \mathbb{R}^{\tau\times5\times3}$, involving both attribute interactions and temporal dependency of the observed electrical sequence. Firstly, we conduct the attribute interaction in each G^t of stack G_τ, which is defined as:

$$\widetilde{\mathbf{V}^t} = \gamma \mathbf{A}^t_{\boldsymbol{\phi}_1} \mathbf{V}^t + (1-\gamma) \mathbf{A}^t_{sim} \mathbf{V}^t, \tag{15}$$

where γ is a hyper parameter; $\mathbf{A}^t_{sim}$ and $\mathbf{A}^t_{\boldsymbol{\phi}_1}$ can be considered as the prior knowledge about the potential interactions between different attributes at time t. Secondly, we stack each $\widetilde{\mathbf{V}^t}$ into $\widetilde{\mathbf{V}_\tau} = \{\widetilde{\mathbf{V}^{t-\tau+1}}, ..., \widetilde{\mathbf{V}^t}\} \in \mathbb{R}^{\tau\times5\times3}$. Thirdly, we feed the $\widetilde{\mathbf{V}_\tau}$ into a temporal residual block shown in Fig. 6(a) to establish the temporal dependency of the working system.

3) **Health estimation**. The third step is to estimate current health condition using the HPN. The HPN receives $\overline{\mathbf{V}_\tau}$ as input and treats the time dimension as feature channels. The output is $\widehat{RUL^t}$. The network architecture is shown in Fig. 6(b).

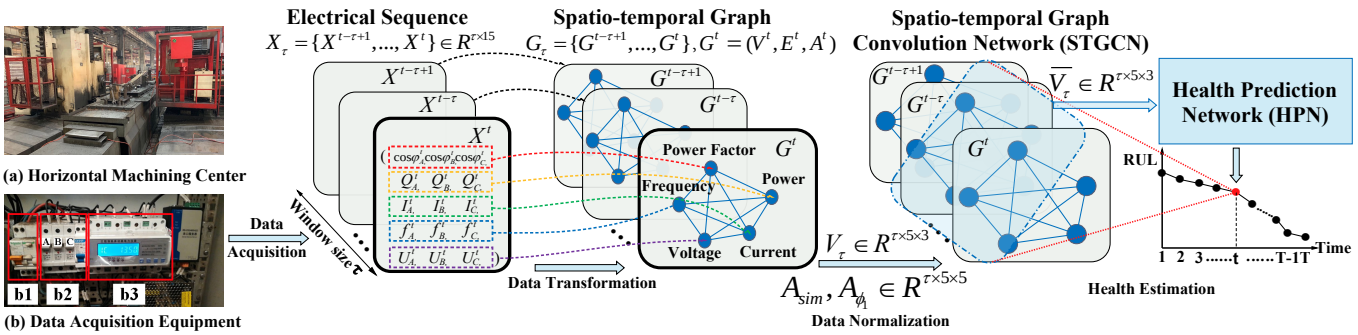

Fig. 5. **Working procedure of the Electrical-STGCN.** Given τ electrical records $\mathbf{X}_\tau$, we first transform this sequence to the constructed spatio-temporal graph $\boldsymbol{G}_\tau = (\boldsymbol{V}_\tau, \boldsymbol{E}_\tau, \mathbf{A}_\tau)$. Then $\boldsymbol{G}_\tau$ is forwarded through the Spatio-Temporal Graph Convolution Network (**STGCN**) to create feature $\overline{\mathbf{V}_\tau}$, which involves both attribute interactions and temporal dependency. After that, the $\overline{\mathbf{V}_\tau}$ is fed into the Health Prediction Network (**HPN**) for current health estimation. **(a)** is the monitored industrial equipment. **(b)** is the power supply circuit of the electrical sensor: **b1** is the leakage protector; **b2** is the three phase fuse; **b3** is the electrical sensor.

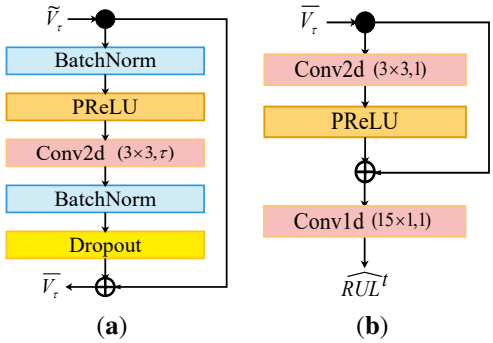

Fig. 6. **(a) Overview of the 5-layer temporal residual block.** Given the $\widetilde{\mathbf{V}_\tau} \in \mathbb{R}^{\tau\times5\times3}$, the output is $\overline{\mathbf{V}_\tau} \in \mathbb{R}^{\tau\times5\times3}$. **(b) The HPN architecture.** The HPN is made up of a residual connection and a convolution layer. Given the $\overline{\mathbf{V}_\tau} \in \mathbb{R}^{\tau\times5\times3}$, the output is $\widehat{RUL}^t$.

Algorithm 1 Pseudo-code of the training procedure

Input: Raw data and the maximum training epoch E_{max}
Output: Electrical-STGCN model

```
Initialize the parameters of the Electrical-STGCN
while e < E_max do
    Transform X_τ to spatio-temporal graph G_τ
    Compute A_sim and A_φ1 using the kernel functions
    Normalize the A_sim, A_φ1, and V_τ
    Conduct attribute interactions by equation (15)
    Conduct temporal dependency (Fig. 6(a))
    Calculate RUL using the HPN (Fig. 6(b))
    Update the parameters of the Electrical-STGCN
    if model convergence then
        Return the Electrical-STGCN
    end if
end while
```

E. Optimization

Training procedure of the Electrical-STGCN is summarized in Algorithm 1. Adam optimizer, a stochastic gradient-based optimization algorithm, is utilized to adjust the learning rate dynamically. The Mean Square Error (MSE) of RUL is used as loss function formulated as:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(RUL - \widehat{RUL}\right)^2, \tag{16}$$

where N is batch size; $\widehat{RUL}$ is the predicted values; and RUL is the ground truth defined as:

$$RUL = \frac{\text{Time to failure} - \text{Current time}}{\text{Time to failure}}. \tag{17}$$

Therefore, the MSE over the training data is calculated and back-propagated to update the parameters in each epoch until model convergence.

IV. Experiments and Analysis

A. Implementation details

In this paper, network parameters are randomly initialized and hyper parameter γ is set to 0.4. The Electrical-STGCN is trained 60 epochs and the batch size is 64. The alarm threshold ξ is empirically set to 0.3 in the working process. If both the ideal RUL and the estimation are below or above ξ, the health condition is considered correct classified. Accuracy, Precision, Recall and F1-Score based on the confusion matrix are used as evaluation metrics. The performances of Electrical-STGCN are achieved by ten-fold cross-validation (i.e., ten maintenance cycles, nine of which are used for training and one for testing in turn). The experiments are all implemented in Pytorch and run on a work station with GeForce RTX 2080 Ti.

B. Exploration study

This section explores the influence of window size τ (i.e., temporal dependency) and kernel functions (i.e., different ways of potential attribute interactions) in detail.

1) Window size τ: In order to explore the impact of window size, we evaluate the performances of the Electrical-STGCN with τ chosen according to the sequence of the powers of 2, to perceive the trend of evaluation metrics over time easily.

As shown in Fig. 7, the Accuracy, Recall, and F1-Score show the similar trend. They reach the best when $\tau = 8$. After that, they decrease from $\tau = 8$ to $\tau = 16$. Then they increase again at $\tau = 32$. Finally, they fall at $\tau = 64$. The Precision reaches the highest when $\tau = 8$ and falls from $\tau = 16$ to $\tau = 64$.

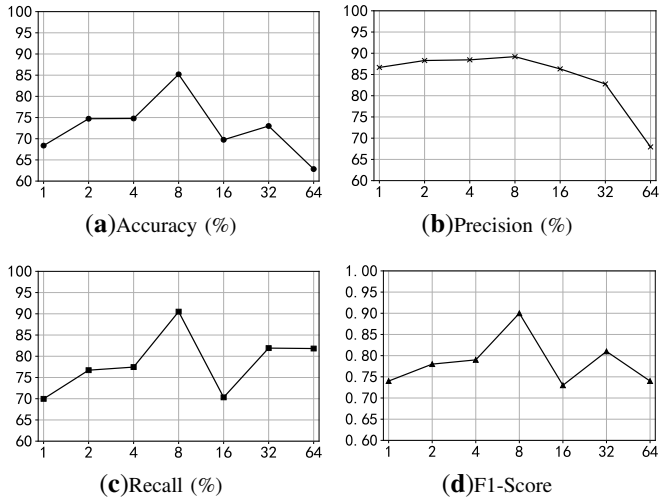

Fig. 7. **Evaluation metrics of the Electrical-STGCN at different window size τ.** The Electrical-STGCN performs the best when $\tau = 8$, i.e., Accuracy= **85.20±0.21**%, Precision= **89.20±0.41**%, Recall= **90.51±0.25**%, and F1-Score= **0.90±0.01**.

The Electrical-STGCN performs the best when $\tau = 8$. We analyze the reason is that $\tau = 8$ might cover the most relevant condition monitoring data for the manufacturing system. Table III shows the model performances on different γ.

TABLE III
ABLATION STUDY ON γ WITH $\tau = 8$.

γ	Accuracy [%]	Precision [%]	Recall [%]	F1-Score
0.2	75.77±1.42	79.53±0.87	91.38±4.17	0.83±0.02
0.3	73.76±0.56	80.97±1.20	86.37±3.51	0.81±0.01
0.4	**85.20±0.21**	**89.20±0.41**	90.51±0.25	**0.90±0.01**
0.5	76.98±0.94	79.89±0.71	**92.67±3.08**	0.84±0.01
0.6	76.14±1.30	81.62±0.89	88.56±4.70	0.82±0.02

2) Kernel functions: To explore the impact of different ways of attribute interactions, we evaluate the performances of the Electrical-STGCN with different kernel functions.

The first candidate is the L_2 norm function, which is used to measure the distance between different vertices defined as:

$$a^t_{L_2(j,k)} = \left\| \boldsymbol{v}^t_j - \boldsymbol{v}^t_k \right\|_2 . \tag{18}$$

The second candidate is Gaussian kernel function applied in [31], which is defined as:

$$a^t_{exp(j,k)} = \frac{exp(-\left\| \boldsymbol{v}^t_j - \boldsymbol{v}^t_k \right\|_2)}{\sigma}. \tag{19}$$

The third candidate is that all diagonal values are set to 1 while the others are set to 0 [32]. It means no attribute interactions, which is utilized as the baseline. The experimental results are shown in Table IV.

Our proposed kernel function improves the Accuracy, Precision, Recall, and F1-Score of the baseline by 7.88%, 5.32%, 2.76%, and 0.07 respectively. Also, it might be more suitable than the L_2 norm and Gaussian kernel function in terms of modeling the electrical working system according to the experimental results.

TABLE IV
THE EFFECT OF DIFFERENT KERNEL FUNCTIONS WITH $\tau = 8$ AND $\gamma = 0.4$.

Kernel Functions	Accuracy [%]	Precision [%]	Recall [%]	F1-Score
Baseline [32]	77.32±1.24	83.88±1.11	87.75±4.43	0.83±0.02
$a^t_{L_2(j,k)}$	84.19±0.28	87.60±0.82	90.63±0.48	0.88±0.01
$a^t_{exp(j,k)}$ [31]	81.37±0.28	85.48±0.65	90.25±0.99	0.87±0.01
$a^t_{sim(j,k)}$ & $a^t_{\phi_1(j,k)}$	**85.20±0.21**	**89.20±0.41**	**90.51±0.25**	**0.90±0.01**

3) Discussion: Fig. 8 is health predictions of the Electrical-STGCN by ten-fold cross-validation. In the working condition (i.e., MC-1 to MC-8), the Electrical-STGCN can successfully warn the anomaly in advance. MC-1 is a bad case that deviates from the ideal value as different workpieces lead to a different distribution of working data. Besides, the RUL in each MC is predicted in a cross-lifecycle manner.

In the off-working condition (i.e., MC-9 to MC-10), no need to consider attribute interactions and temporal dependency, the Electrical-STGCN performs well in MC-9. However, MC-10 is a failure case, which left no time for maintenance. Therefore, how to improve the performance in the off-working condition to warn the anomaly in advance is an important point in our future work. Another limitation of the Electrical-STGCN is the number of the collected attributes. Because the model is graph-based, it is more suitable for the case with enough attributes since there are rich information interactions.

C. Comparison with other ML algorithms

SVM, K-Means, RFs, LSTM, GRU, ConvLSTM, and TCN are compared with the Electrical-STGCN. Accuracy, Precision, Recall, and F1-Score based on the confusion matrix are achieved through ten-fold cross-validation. Furthermore, the Wilcoxon signed-rank test with significance level 0.05 is utilized to compare the significant differences between the Electrical-STGCN and other competitors.

In the experiment, Gaussian kernel is employed in the SVM; the cluster number of the K-Means is set to 2; the number of estimators in the RFs is set to 20; the dimension of hidden state $\mathbf{h}_t$ and window size τ in the LSTM and GRU are set to 16 and 8, respectively; the kernel size of the ConvLSTM is set to 3. Experimental results and Wilcoxon sign-ranked test are shown in Table V.

TABLE V
COMPARISON WITH OTHER ML ALGORITHMS IN TERMS OF WILCOXON SIGNED-RANK TEST.

Competitors	Accuracy [%]	Precision [%]	Recall [%]	F1-Score
SVM [20]	49.85±3.56(+)	74.92±15.79(≡)	32.42±9.93(+)	0.40±0.10(+)
K-Means [23]	63.69±4.28(+)	80.15±4.05(≡)	67.47±3.87(+)	0.72±0.02(+)
RFs [25]	76.57±1.23(+)	85.15±1.19(≡)	84.92±5.17(≡)	0.81±0.02(+)
LSTM [27]	77.25±2.02(≡)	88.48±1.70(≡)	81.44±6.05(≡)	0.82±0.03(≡)
GRU [28]	77.98±1.66(+)	87.81±1.41(≡)	79.06±5.59(≡)	0.81±0.02(+)
ConvLSTM [29]	77.60±1.11(+)	83.73±1.66(≡)	89.67±2.75(≡)	0.84±0.01(+)
TCN [30]	64.43±2.62(+)	82.99±2.65(≡)	67.51±9.55(≡)	0.69±0.04(+)
E-STGCN(ours)	**85.20±0.21**	**89.20±0.41**	**90.51±0.25**	**0.90±0.01**

$^+$ means that Electrical-STGCN is significantly better than other competitors.
$^\equiv$ means that there is no significant difference between Electrical-STGCN and others based on Wilcoxon signed-test with the significance level of 0.05.

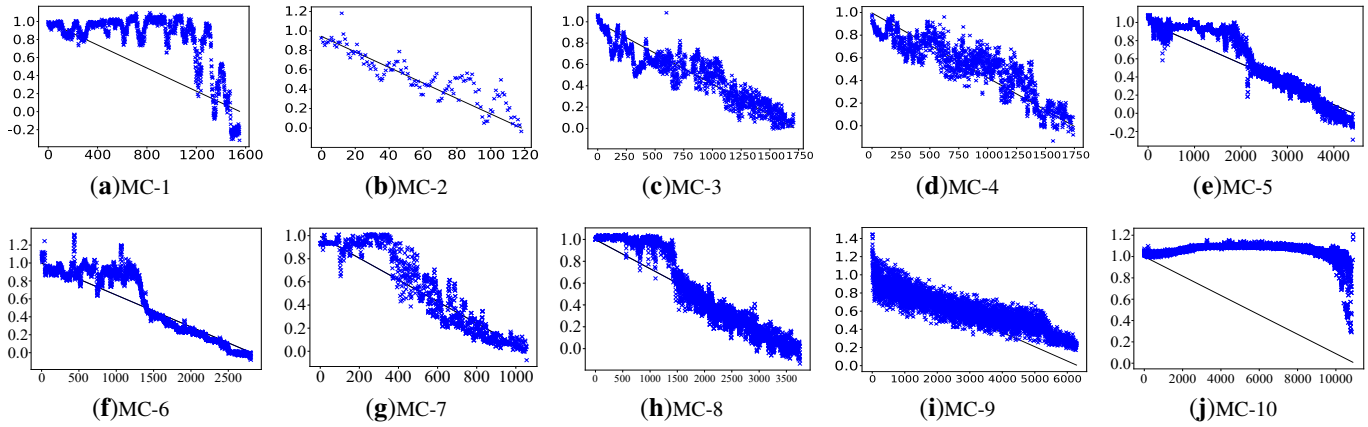

Fig. 8. **Health condition of each maintenance cycle (MC) predicted by the Electrical-STGCN.** Horizontal axis is time (s), and vertical axis is RUL. MC-1 to MC-8 are in the working condition while MC-9 to MC-10 are in the off-working condition where anomalies occur randomly. The black line is the ideal RUL. Blue dots are predictions.

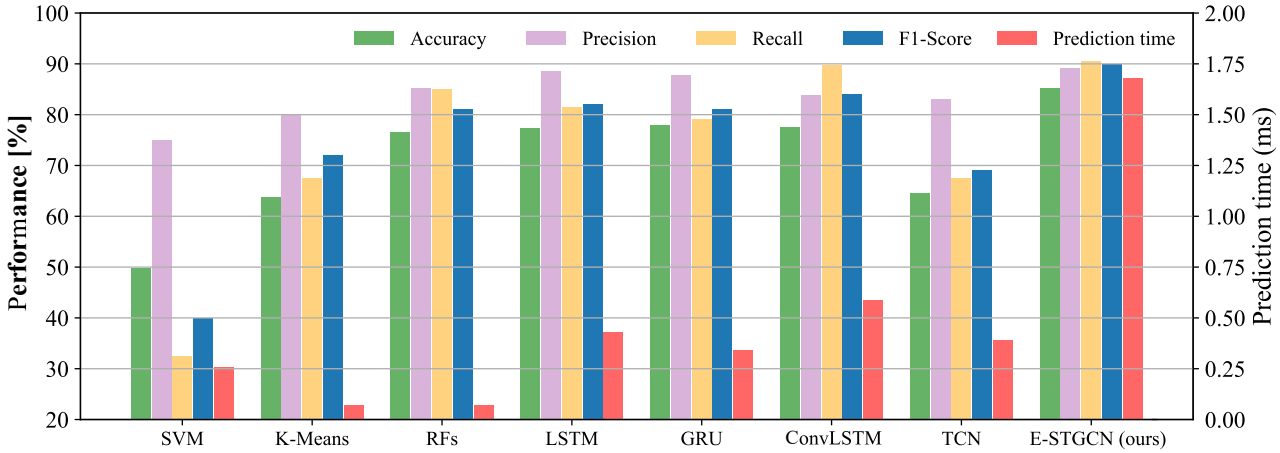

Fig. 9. **Comparison with different ML algorithms in terms of Accuracy, Precision, Recall, F1-Score, and Prediction time.** We conduct the ML algorithms on the testing data and calculate the average processing time for each record.

Generally, by considering temporal dependency of the working system, the RNN-based models (i.e., LSTM, GRU, and ConvLSTM) perform better than the SVM, K-Means, and RFs. Specifically, the RFs performs better than SVM and K-Means because it generates 20 weak classifiers and aggregates them together to form better results.

Considering both attribute interactions and temporal dependency, the Electrical-STGCN performs the best. It dominates the SVM and K-Means in terms of Accuracy, Precision, and F1-Score. It also dominates RFs, GRU, TCN, and ConvLSTM in terms of Accuracy and F1-Score. However, there is no significant difference with LSTM. The LSTM performs well in short maintenance cycle while the Electrical-STGCN performs better in long maintenance cycle. The reason is that Electrical-STGCN considers attribute interactions in addition to temporal dependency, which allows to model the working system better.

As shown in Fig. 9, it takes the Electrical-STGCN 1.68 ms to estimate the current RUL. Although the Electrical-STGCN can not outperform the other competitors in terms of prediction time, it can run in real-time. The computational cost mainly depends on conducting the adjacency matrices of the electrical data sequence.

V. Conclusion and Future Work

This paper proposes an Electrical-STGCN for intelligent PdM of the manufacturing system. During the experiments, we have seen how different window size and kernel functions affect the performances in detail. The effectiveness of the Electrical-STGCN is demonstrated by real-world cases.

From this work, we made some key observations. Firstly, there exists the best window size that covers the most relevant condition monitoring data of the working system. By utilizing the best temporal dependency, we can establish reliable model for the system. In addition, there are potential transfer relationships between the collected attributes. It is crucial to establish attribute dependencies to achieve better performances.

The future work will mainly concentrate on two aspects. First, it might be interesting to employ graph attention mechanism in the Electrical-STGCN to assign higher weights to the attributes that contribute the most to final results automatically. That might be helpful to capture anomalies in different working conditions. Second, it is worth to search for a method to map the attributes to higher dimension. That might improve the generality of the Electrical-STGCN.

REFERENCES

[1] W. Zhang, D. Yang, Y. Xu, X. Huang, J. Zhang, and M. Gidlund, "Deephealth: A self-attention based method for instant intelligent predictive maintenance in industrial internet of things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5461–5473, 2021.

[2] M. S. S. Garmaroodi, F. Farivar, M. S. Haghighi, M. A. Shoorehdeli, and A. Jolfaei, "Detection of anomalies in industrial iot systems by data mining: Study of christ osmotron water purification system," *IEEE Internet of Things Journal*, 2020.

[3] G. A. Susto, A. Beghi, and C. De Luca, "A predictive maintenance system for epitaxy processes based on filtering and prediction techniques," *IEEE Transactions on Semiconductor Manufacturing*, vol. 25, no. 4, pp. 638–649, 2012.

[4] R. Y. Zhong, X. Xu, E. Klotz, and S. T. Newman, "Intelligent manufacturing in the context of industry 4.0: a review," *Engineering*, vol. 3, no. 5, pp. 616–630, 2017.

[5] S. Ayvaz and K. Alpay, "Predictive maintenance system for production lines in manufacturing: A machine learning approach using iot data in real-time," *Expert Systems with Applications*, vol. 173, p. 114598, 2021.

[6] J. C. Cheng, W. Chen, K. Chen, and Q. Wang, "Data-driven predictive maintenance planning framework for mep components based on bim and iot using machine learning algorithms," *Automation in Construction*, vol. 112, p. 103087, 2020.

[7] H.-C. Chen, K. T. Putra, S.-S. Tseng, C.-L. Chen, and J. C.-W. Lin, "A spatiotemporal data compression approach with low transmission cost and high data fidelity for an air quality monitoring system," *Future Generation Computer Systems*, vol. 108, pp. 488–500, 2020.

[8] J. C.-W. Lin, G. Srivastava, Y. Zhang, Y. Djenouri, and M. Aloqaily, "Privacy-preserving multiobjective sanitization model in 6g iot environments," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5340–5349, 2020.

[9] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical systems and signal processing*, vol. 20, no. 7, pp. 1483–1510, 2006.

[10] X.-S. Si, W. Wang, C.-H. Hu, and D.-H. Zhou, "Remaining useful life estimation–a review on the statistical data driven approaches," *European journal of operational research*, vol. 213, no. 1, pp. 1–14, 2011.

[11] A. Mosallam, K. Medjaher, and N. Zerhouni, "Data-driven prognostic method based on bayesian approaches for direct remaining useful life prediction," *Journal of Intelligent Manufacturing*, vol. 27, no. 5, pp. 1037–1048, 2016.

[12] H. Liao, W. Zhao, and H. Guo, "Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model," in *Annual Reliability and Maintainability Symposium, 2006.* IEEE, 2006, pp. 127–132.

[13] J. M. van Noortwijk, "A survey of the application of gamma processes in maintenance," *Reliability Engineering & System Safety*, vol. 94, no. 1, pp. 2–21, 2009.

[14] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 2017, pp. 1965–1972.

[15] T. D. Batzel and D. C. Swanson, "Prognostic health management of aircraft power generators," *IEEE Transactions on Aerospace and electronic systems*, vol. 45, no. 2, pp. 473–482, 2009.

[16] D. Banjevic, "Remaining useful life in theory and practice," *Metrika*, vol. 69, no. 2-3, pp. 337–349, 2009.

[17] A. Kumar, R. B. Chinnam, and F. Tseng, "An hmm and polynomial regression based approach for remaining useful life and health state estimation of cutting tools," *Computers & Industrial Engineering*, vol. 128, pp. 1008–1014, 2019.

[18] R. Li, W. J. Verhagen, and R. Curran, "A comparative study of data-driven prognostic approaches: Stochastic and statistical models," in *2018 IEEE International Conference on Prognostics and Health Management*. IEEE, 2018, pp. 1–8.

[19] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM transactions on intelligent systems and technology*, vol. 2, no. 3, pp. 1–27, 2011.

[20] G. A. Susto, S. McLoone, D. Pagano, A. Schirru, S. Pampuri, and A. Beghi, "Prediction of integral type failures in semiconductor manufacturing through classification methods," in *2013 IEEE 18th Conference on Emerging Technologies & Factory Automation*. IEEE, 2013, pp. 1–4.

[21] H. Li, D. Parikh, Q. He, B. Qian, Z. Li, D. Fang, and A. Hampapur, "Improving rail network velocity: A machine learning approach to predictive maintenance," *Transportation Research Part C: Emerging Technologies*, vol. 45, pp. 17–26, 2014.

[22] K. Dhalmahapatra, R. Shingade, H. Mahajan, A. Verma, and J. Maiti, "Decision support system for safety improvement: an approach using multiple correspondence analysis, t-sne algorithm and k-means clustering," *Computers & Industrial Engineering*, vol. 128, pp. 277–289, 2019.

[23] E. Uhlmann, R. P. Pontes, C. Geisert, and E. Hohwieler, "Cluster identification of sensor data for predictive maintenance in a selective laser melting machine tool," *Procedia Manufacturing*, vol. 24, pp. 60–65, 2018.

[24] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*. Springer, 2000, pp. 1–15.

[25] M. Canizo, E. Onieva, A. Conde, S. Charramendieta, and S. Trujillo, "Real-time predictive maintenance for wind turbines using big data frameworks," in *2017 IEEE International Conference on Prognostics and Health Management*. IEEE, 2017, pp. 70–77.

[26] P. Li and O. Niggemann, "A nonconvex archetypal analysis for one-class classification based anomaly detection in cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 6429–6437, 2020.

[27] O. Aydin and S. Guldamlasioglu, "Using lstm networks to predict engine condition on large scale data processing framework," in *2017 4th International Conference on Electrical and Electronic Engineering*. IEEE, 2017, pp. 281–285.

[28] Y.-W. Lu, C.-Y. Hsu, and K.-C. Huang, "An autoencoder gated recurrent unit for remaining useful life prediction," *Processes*, vol. 8, no. 9, p. 1155, 2020.

[29] A. Essien and C. Giannetti, "A deep learning model for smart manufacturing using convolutional lstm neural network autoencoders," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6069–6078, 2020.

[30] Y. Wang, L. Deng, L. Zheng, and R. X. Gao, "Temporal convolutional network with soft thresholding and attention mechanism for machinery prognostics," *Journal of Manufacturing Systems*, vol. 60, pp. 512–526, 2021.

[31] Y. Zhang, Y. Li, X. Wei, and L. Jia, "Adaptive spatio-temporal graph convolutional neural network for remaining useful life estimation," in *2020 International joint conference on neural networks*. IEEE, 2020, pp. 1–7.

[32] M. Wang, Y. Li, Y. Zhang, and L. Jia, "Spatio-temporal graph convolutional neural network for remaining useful life estimation of aircraft engines," *Aerospace Systems*, vol. 4, no. 1, pp. 29–36, 2021.

[33] E. Ramasso, M. Rombaut, and N. Zerhouni, "Joint prediction of continuous and discrete states in time-series based on belief functions," *IEEE transactions on cybernetics*, vol. 43, no. 1, pp. 37–50, 2012.

[34] F. Chung and S.-T. Yau, "Discrete green's functions," *Journal of Combinatorial Theory, Series A*, vol. 91, no. 1-2, pp. 191–214, 2000.