This is the Pre-Published Version.

**Proceedings of the ASME 2020
International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
IDETC/CIE2021
August 17-20, 2021, Virtual, Online**

DETC2021-XXXX

PREDICTING THE MATERIAL REMOVAL RATE IN CHEMICAL MECHANICAL PLANARIZATION PROCESS: A HYPERGRAPH NEURAL NETWORK-BASED APPROACH

Liqiao Xia[1] Pai Zheng*[1] Chao Liu[1]

[1]Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China

ABSTRACT

Material removal rate (MRR) plays a critical role in the operation of chemical mechanical planarization (CMP) process in the semiconductor industry. To date, many physics-based and data-driven approaches have been proposed to predict the MRR. Nevertheless, most of the existing methodologies neglect the potential source of its well-organized and underlying equipment structure containing interaction mechanisms among different components. To address its limitation, this paper proposes a novel hypergraph neural network-based approach for predicting the MRR in CMP. Two main scientific contributions are presented in this work: 1) establishing a generic modeling technique to construct the complex equipment knowledge graph with a hypergraph form base on the comprehensive understanding and analysis of equipment structure and mechanism, and 2) proposing a novel prediction method by combining the Recurrent Neural Network based model and the Hypergraph Neural Network to learn the complex data correlation and high-order representation base on the Spatio-temporal equipment hypergraph. To validate the proposed approach, a case study is conducted based on an open-source dataset. The experimental results prove that the proposed model can capture the hidden data correlation effectively. It is also envisioned that the proposed approach has great potentials to be applied in other similar smart manufacturing scenarios.

Keywords: material removal rate, graph neural network, recurrent neural network, hypergraph, chemical mechanical planarization

1. INTRODUCTION

Chemical mechanical planarization (CMP) is widely used in semiconductor manufacturing to planarize globally. A classical CMP system includes a rotating table used to hold a polishing pad as well as a replaceable polishing pad stuck to the rotating table. Above that, a rotating wafer carrier is used to push a wafer from up to down and the wafer is also held by the polishing pad on the other side. Additionally, a slurry dispenser and a rotating dresser involve in the polishing process on the polishing pad. During the process of CMP, the wafer is pushed against the polishing pad and the wafer carrier above and a polishing pad below the wafer are spinning in the same direction. Such operation removes the film on the exterior of the wafer which is created by the chemical reaction initially[1].

Material removal rate (MRR) is one of the essential metrics to measure the quality of CMP since defect and depression generated on wafers material will increase the fault rate of CMP[2]. Traditionally, research focuses on the study of the effect of components[3] and manufacturing environment[4] in CMP and how they affect the MRR. Moreover, mathematical approaches try to fit a curve to predict the MRR[5] or establish a mathematics model to simulate the manufacturing process[6]. Nevertheless, with an increasing number of devices to collect multi-modal data in manufacturing and more computation power, machine learning and deep learning approaches are increasingly implemented to predict the MRR.

The structural information contained in the equipment will benefit the MRR prediction model. Most equipment including CMP owns a pre-defined and clear mechanism which indicates its corresponding connections among the inner components. On the one hand, to our best knowledge, previous approaches fail to utilize the structure of the equipment in their data-driven prediction model. On the other hand, though there are many knowledge graphs-based or ontology-based applications in industrial scenarios before, these graphs contain different kinds of nodes and corresponding edges. For example, the edge attribute typically includes ‘is part of’, ‘lead to’, ’has a function’, and so on. These edges and the corresponding graph serve as

other purposes, like Q&A system and the smart product-service system, rather than modeling the impact among components in the graph form.

Though few industrial knowledge graphs have been developed to present the impact among components of equipment so far, many research works have proposed enterprise knowledge graphs and reasoning in other domains. Establishing enterprise knowledge graphs usually requires domain-specific knowledge and expertise. However, the reasoning methods are similar: rules-based, structure-based, and deep learning methods, among which deep learning methods are most popular. GNNs, deep learning approaches that specialize in network structure, have been introduced to tackle the problem with graph form[7]. The more advanced models derived from GNN, such as GCN and Graph-SAGE, have been implemented in many other scenarios. For example, it can be used for text classification, relation extraction, reasoning, as well as object detection, and semantic segmentation in the computer vision domain[8]. It is noted that the industrial knowledge graph is extremely different from the other knowledge graphs and therefore should be specialized.

Aiming to fill this research gap, this paper proposes an industrial graph to represent the structural knowledge, and a time-domain hypergraph model for prediction. The rest of this paper is organized as follows. Section 2 reviews the related works on industrial knowledge graph categories and the approaches of the MRR prediction model. Section 3 illustrates the pre-requisite information of the GCN model and hypergraph. Section 4 presents a case study and discusses the experimental results. Section 5 provides the conclusions and future work.

2. Related Works

The previous MRR prediction approaches can be divided into physics-based and data-driven. One of the most popular physics-based approaches is the Preston equation[1] which indicates $MRR = K_p P^{\alpha} V^{\beta}$, where P represents the downward pressure push to a wafer, V represents the rotating speed, $K_p$is the Preston coefficient. The majority of physics-based methods are proposed based on the Preston equation. For example, adding contact stress, relative velocity, and chemical reaction rate into the Preston coefficient[6]. Other research takes the size, concentration, distribution of particles, slurry flow rate, polishing pad surface topography into consideration[9]. The limitation of physics-based approaches is that it requires prior assumptions. Another mainstream approach is data-driven based, among which machine learning and statistics method is a common way. Previous researchers implement the nonlinear Bayesian model[10] and the decision tree-based model[11]. Recently, with the rapid development of deep learning, increasing research apply deep learning methods to predict the MRR. Base on the fundamental structure of a deep neural network, a deep belief network is proposed to predict the MRR[12]. Furthermore, an adaptive method[13] or an adaptive neuro-fuzzy inference[14] is combined with a classical neural network to improve the performance. While the above approaches neglect the structural information of equipment, this paper should focus on how to establish an industrial knowledge graph.

For the industrial knowledge graph, especially in the equipment domain, recent works have established industrial knowledge graph in different understand and perspectives. Some recent works digitize part or all of the equipment information or digitalize the system working process, and even the entire production process, and link the data in different vertical fields to construct the corresponding knowledge graph. This method can help operators to find potential key components and overall modification plans. A very straightforward way is transforming the working process into a knowledge graph[15] or disassembling the components as nodes in the knowledge graph[16]. More advanced, some work builds several knowledge graphs to satisfy their requirement[17]. Moreover, an event graph is generated to simulate and understand the manufacturing process and represent the event logic with a clear description in graph form[18].

While the above methods cannot fully reflect the operation of the equipment and the interaction between components especially for CMP. Therefore, this paper introduces a hypergraph to establish the graph. The hypergraph has been first introduced as a propagation process on the hypergraph structure in 2008[19]. Later it had been implemented in other domains, such as video[20] and image[21]. To increase the effectiveness of the hypergraph structure, other works focus on the weight of hyperedge. By improving the algorithm, l_2 has been used in the model to learn optimal hyperedge weight[22]. Or assuming that highly correlated hyperedges should have similar weight[23]. Recently, a hypergraph has been implemented with a neural network that combines the hypergraph with a graph convolutional network to consider the complex data correlation[24].

Because many unstructured data can only be represented in the graph form, research work has paid more attention to applying neural networks to the graph structure. For example, tackling Spatio-temporal graphs is applying RNN into graph networks [25]. Another popular method is GCN, which implements a convolutional layer into a graph network[8].

In the industrial scenario, when facing the prediction problem(e.g. RUL estimation), there are already mature solutions based on deep learning and machine learning[26]. While recent research work focuses on incorporating the semantics of the problem into the structure of deep learning models by using ontologies. For example, building the deep learning network by the ontology design and stacking the LSTM module to learn temporal dependencies[27]. Another example is to simulate equipment as a graph and its nodes represent the sense data, it constructs the graph by fully connected between every two nodes(sense data), and the edge reflects the dependencies of these two connected sense data[28]. The above research work attempts to solve different tasks base on the equipment graph, while they fail to model the interaction between different components and their corresponding feature explicitly.

Overall, the above research mainly focuses on predicting the MRR with physics-based and data-driven approaches without using equipment structure and industrial knowledge graph for other tasks, while few work, components, or modules are treated as nodes has attempted to implement a knowledge graph with equipment structure for MRR prediction. To address this research gap, a novel hypergraph neural network-based approach is proposed.

3. Methodology

In this section, the knowledge graph data model of the industrial domain is illustrated firstly. It provides a comprehensive framework that stores and manages the various manufacturing data systematically and graphically. Based on the semantic understanding of specific equipment cooperate with this knowledge graph model, one can summarize the type of module, its corresponding specific node, and describe the affiliation between them. Subsequently, the technology of hypergraph construction will be introduced which tackles the multi-impact problem. By using this technique to determine the relationship between nodes and edges, the model becomes more explanatory and exploratory.

3.1 Knowledge Graph data model

Different from the traditional knowledge graph, components or modules are treated as nodes in the equipment knowledge graph model, and the interaction between them serves as links. The nodes in the graph which correspond to the components or modules assemble the physical equipment system and each of them will contain one or many sensor data. The related information of the equipment component is well organized in ontology or tree diagram. Besides, by analyzing the equipment operation mode and internal interaction, the links between different modules can be determined.

The first step is building a graph to illustrate the affiliation of data and modules. Each component or module will contain relevant data such as sensor data and manufacturing logs. Those data serve as a sub-node under the up-level node (its corresponding module or component) and those sub-nodes will store their data as their node's features. An initial graph structure is shown in Fig 1.

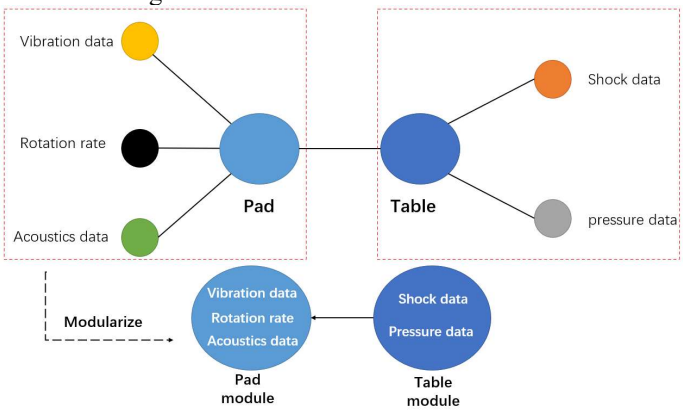

Fig 1. The example of a module and its corresponding attributes

Take part of the equipment as an example, as shown in the upper of Fig 1, Pad and Table are adjacent components connected physically and there is a push force from table module to pad module, so they will link to each other. On the other hand, three features (Vibration data, rotation rate, acoustics data) are collected from Pad, and two features (shock data and pressure data) are collected from Table. Therefore, those data link to their up-level nodes respectively.

The second step is to generate the module by the relevant nodes. As shown in the lower part of Fig 1, the blue background point represents one single module, setting them as one node which contains the information of its relevant feature. In the mechanism of CMP, a push force is acting on the pad module from the table module, therefore, there is a directed link from the table module to the pad module.

3.2 Hypergraph construction

Setting the module as a node, the connection among nodes can be established by analyzing the assembly or CAD. While in most scenarios, the interaction always occurs among more than two modules. Under this circumstance, it is hard to determine what is the exact mathematical expression or weight among different edges in the same interaction due to the limitation of data and prior knowledge available. For example, on the left side of Fig 2, an upward force pushes the bottom of the Pad, and the Pad holds a dresser and a wafer. In this scenario, there will be interaction force among dresser, pad, and wafer. But it is very difficult to clarify the difference between the dresser and pad and the one between wafer and pad because the sense data is the result of their interaction of these three modules. Under this circumstance, hyperedge can be implemented to tackle this problem. By utilizing hyperedge, the edge between pad and wafer and the edge between pad and dresser merge as one hyperedge. As shown on the right side of Fig 2, these two edges merge as one hyperedge, linking from the pad module to the wafer module and dresser module. The directed hyperedge comes from the mechanism that an upward force pushes the pad, thereby a force will interact from pad to dresser and wafer as shown in Fig 2.

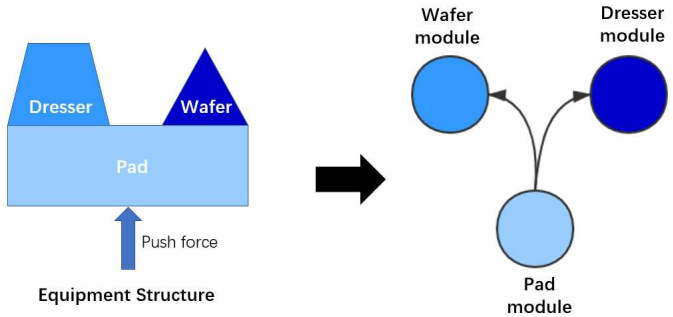

Fig 2. Convert equipment structure into a hypergraph

Under this hyperedge schema, it is not necessary to define the exact relationship or edge between two nodes, which is the ambiguous and difficult part in the knowledge graph construction due to the limited data or prior knowledge. For example, for the module of the pad, wafer, and dresser, each of them will have the corresponding sense-data, such as Vibration data, Acoustics data. However, it is difficult to ignore the influence of the associated modules and obtain the interaction

relationship between two modules through the sensor data. Taking the above example, there is an estimated error if calculating the interaction between the wafer module and pad module by their belonging sense data but without the dresser one. Therefore, all the relevant nodes share same hyperedge which represents the interaction among them.

3.3 Prediction Algorithm

The prediction algorithm aims to consider the time-domain feature and the graph structure by adopting convolution in the hypergraph and RNN-based model in the time domain feature.

Setting a hypergraph $G = (V, \varepsilon, \Delta)$ where V represents n nodes and ε represents k edges. Δ is a Laplacian n*n positive semi-definite matrix, with the mathematics eigendecomposition $\Delta = \Phi\Lambda\Phi^T$ to obtain the orthonormal eigenvectors Φ and corresponding diagonal matrix Λ which contains the non-negative eigenvalues. Consequently, applying the Fourier transform to the spectral convolution of signal x and filter g can be denoted as

$$g \star x = \Phi\left((\Phi^{\mathrm{T}} g) \odot (\Phi^{\mathrm{T}} x)\right) = \Phi g(\Lambda)\Phi^{\mathrm{T}} x, \quad (1.)$$

where $\odot$ is the element-wise Hadamard product and $g(\Lambda)$ is the function of Fourier coefficients. While the forward computation is large and the inverse Fourier transfer also takes $O(n^2)$ time complexity. This paper utilizes the Chebyshev expansion as the polynomial to get the approximate result.

$$g \star x \approx \sum_{k=0}^{K} \theta_k T_k(\tilde{\Delta}) x, \quad (2.)$$

where $T_k(\tilde{\Delta})$ is the Chebyshev polynomial of order K. Since the Laplacian in hypergraph can fully represent the features' high order interaction, it sets the K=1 to constrict the order of convolutional procession. With the recommended Λ[31], the convolutional operation equation as be simply written as

$$g \star x \approx \theta_0 x - \theta_1 D^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} x, \quad (3.),$$

where θ_0 and θ_1 are the weights of the filter, D_v is the vertex degree matrix and D_e is the hyperedge degree, H is the hypergraph adjacent matrix. Here this paper simplifies the parameter θ from two unknown parameters to one unknown parameter by the following setting

$$\begin{cases} \theta_1 = -\frac{1}{2}\theta \\ \theta_0 = \frac{1}{2}\theta D_v^{-1/2} H D_e^{-1} H^T D_v^{-1/2} \end{cases}$$

Putting the transformation into (3):

$$g \star x \approx \frac{1}{2}\theta D_v^{-\frac{1}{2}} H(W + I) D_e^{-1} H^T D_v^{-\frac{1}{2}} x, \quad (4.)$$

$$\approx \theta D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} x$$

where I is the identity matrix, so W+I approximately equals to W, and 1/2 can also around 1.

Since this transfer calculates in the spectral domain, it needs to apply the convolution filter Θ to inverse transform to the spatial domain.

$$Y = D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} X\Theta, \quad (5.)$$

where $W=\mathrm{diag}(w_1, w_2, \ldots, w_n)$, $\Theta \in \mathbb{R}^{C_1 * C_2}$, C_1 and C_2 are the feature dimension before and after convolution.

Besides, establishing a LSTM model for each time series feature $X_t^n = \{x_0^n, \ldots, x_t^n\}$. The mathematical algorithm of the LSTM model is briefly recalled here because this model is one of the backbones of the whole algorithm.

In the first step, the forget gate function f_t need to be calculated which determine how much information should be saved from the last cell into the next cell:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t^n] + b_f\right), \quad (6.)$$

where W_f is the weight matrix of forgetting gate function, b_f is the bias value and h_t is the hidden state in time *t*.

Opposite to forget gate f_t, an input gate i_t determine how much information need to be updated and V_t^T is a state to be updated:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t^n] + b_i), \quad (7.)$$

$$V_t^T = tanh(W_c \cdot [h_{t-1}, x_t^n] + b_c), \quad (8.)$$

combining the gates and state obtained above, a new state value can be calculated as followed:

$$V_t = f_t * C_{t-1} + i_t * V_t^T. \quad (9.)$$

After the calculation of the new state C_t, the hidden state h_t also need to update for the next iteration: the o_t represents updated information and the final result h_t in equation (11):

$$o_t = \sigma(W_o[h_{t-1}, x_t^n] + b_o), \quad (10.)$$

$$h_t = o_t * \tanh(V_t), \quad (11.)$$

with the equation above, we can get the output state value h_t of timestamp t from the input value x_t^n.

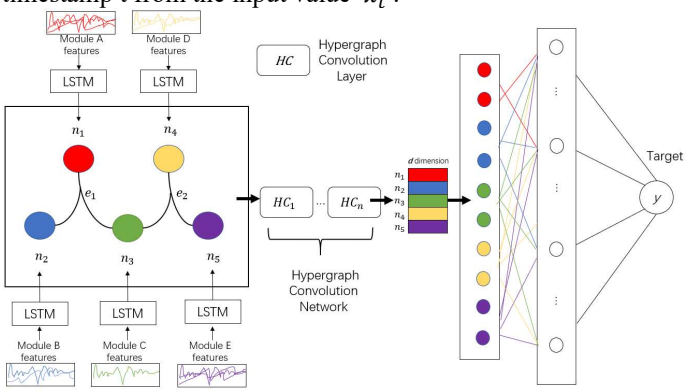

Fig 3. The algorithm structure of hypergraph neural network for prediction

The output state value h_t also represents the node-level feature in the hypergraph. A hypergraph G can be presented in matrix form H by $|V| \times |\varepsilon|$ with the following definition:

$$h(v, e) = \begin{cases} 1, if\ v \in e \\ 0, if\ v \notin e \end{cases}$$

Hence, the iteration equation can be rewritten from (5) to

$$X_t^{l+1} = \sigma\left(D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} X_t^l \Theta\right), \quad (12.)$$

where $X_t^0 = X_t$ and σ denotes the sigmoid function. This hypergraph iteration equation utilizes the core idea of spectral convolution on the graph structure. As shown in Fig 4, the hypergraph neural network achieves node-edge-node

transformation so that it can extract the high order of the features. Initially, X multiply by H^T can transform the node level feature into a hyperedge feature representing gathering accorded to the hyperedge. Subsequently, by multiplying matrix H, the final output refines node feature is generated which means aggregated their related hyperedge feature. In (12), D_v and D_e have a normalization function. Therefore, by utilizing the node-hyperedge-node mechanism, the hypergraph neural network extracts the high-order feature efficiently.

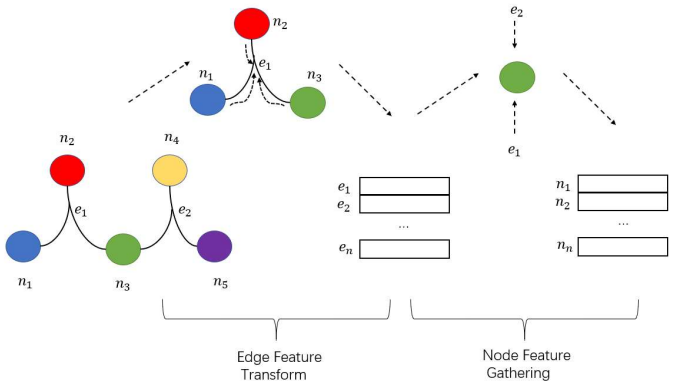

Fig 4. The Schematic diagram of how node embedding generates

The refine output from the hypergraph neural network $\tilde{V} = \{\widetilde{v_0}, \dots, \widetilde{v_n}\}$ reads out as a graph level output by concatenating them. With this concatenation, the graph level output represents as $\tilde{V}^T = [\widetilde{v_0}, \dots, \widetilde{v_n}]$. The final estimated value can be calculated through a fully connected layer as follow:

$$v_{hidde} = g(W * \tilde{v}^T + b_{h1}) \quad (13)$$

$$v_{output} = g(W * v_{hidde} + b_{h2}) \quad (14)$$

Where g is the non-linear active function.

$$L = \frac{1}{n}\sum_{i}^{n}\left(y_{output} - y_{true}\right)^2 \quad (15)$$

4. Case Study

To empirically demonstrate the proposed approach, an illustrative example of a wafer chemical-mechanical planarization is adopted. CMP implements to remove the surface material of the wafer. Combing with the corrosive chemical slurry, polishing pad, and retaining ring to polish the residue of the wafer in the CMP.

As shown in Fig 5. The wafer is installed in the carrier on a polishing pad backing film and the retaining ring on the wafer carrier maintains the wafer in a correct horizontal position. Both the wafer carrier and the polishing pad are rotating during the polishing process and a downward force is acting on the carrier to push the wafer against the polishing pad. Meanwhile, the slurry outflow from the slurry dispenser includes chemical components and abrasive particles. This CMP data is collected from the open data source from the competition of PHMS 2016. This data aims to predict the material removal rate of each processing.

4.1 Data Description

The dataset contains multiple sensory signals collected from a CMP tool that removes the material from wafers. This dataset contains 14 features and their corresponding wafer id and MRR. The 14 features mainly include the usage of the polish-pad backing film, dresser, polishing table, dresser table, wafer carrier sheet, the flow rate of slurry, and the pressure of different components. The number of the total dataset is 376859 and corresponding to 1166 wafers records (a distinct wafer has many timestamp records and one single MRR) and this experiment split 75% of dataset as training dataset and the rest as testing dataset. The predictive model was trained on the training dataset and then validated on the test datasets. The test datasets provide an unbiased evaluation of the model fit on the training dataset.

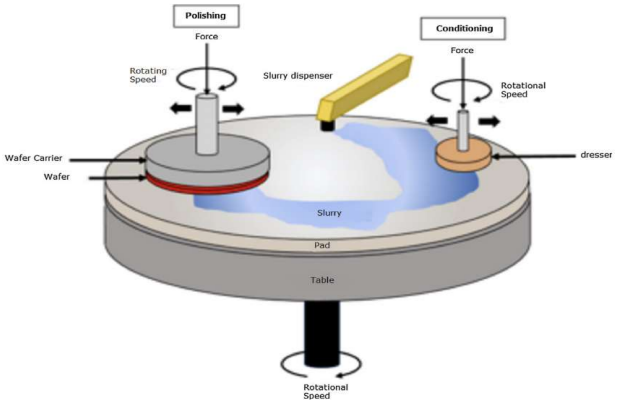

Fig 5. The Schematic diagram of CMP

4.2 Hypergraph construction

The specific CMP equipment, as shown in Fig 6, can be divided into five modules base on its structure and interaction.

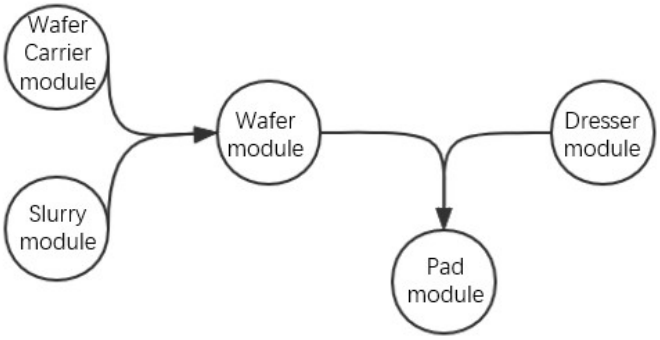

Fig 6. The hypergraph of the CMP structure of five modules

First of all, according to the mechanism of CMP, a downward force is applied to the wafer carrier to push the wafer toward the pad. Therefore, a directed edge links from the wafer carrier module to the wafer module. Besides, the wafer materials were passivated and etched by the slurry chemicals, which represents the slurry module has an initial impact on the wafer module. Meanwhile, the sense data of the wafer module is influenced by both the wafer carrier module and slurry, it is difficult to estimate how each of them affects the wafer module separately. Therefore, setting them as a hyperedge to represent the collaborative influence relationship. Similarly, the wafer pushes against the pad, which reflects the directed edge from the wafer module to the pad module. Besides, the dresser used to roughen the pad surface which also represents a directed edge from the

dresser to pad. Both the wafer and the dresser connect vertically with the pad have different initiative forces so a hyperedge is used to represent this multi-edge relationship in equipment.
After analyzing the relationship between different modules, a hypergraph is generated. Each of the modules contains one or more features.

TABLE I
FEATURES MAPPING CATEGORY EXAMPLE

Feature	Category
A usage measure of the polish-pad backing film	Pad module
A usage measure of the dresser	Dresser module
Pressure related to wafer placement	Wafer module
The flow rate of slurry	Slurry module
A usage measure of wafer carrier flexible sheet	Wafer carrier module
…	…

4.3 Average removal rate prediction

Transferring the specific CMP hypergraph structure as a hyper matrix *H*.

$$H = \begin{array}{c} \begin{matrix} e_1 & e_2 \end{matrix} \\ \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix} \end{array} \begin{array}{l} \text{Wafer carrier module} \\ \text{Slurry module} \\ \text{Wafer module} \\ \text{Pad module} \\ \text{Dresser module} \end{array}$$

$$Where\ H_{ij} = \begin{cases} 1, if\ e_j\ links\ with\ module\ i \\ 0, \quad otherwise \end{cases}$$

Fig 7. The hyper matrix of the CMP hypergraph structure

By analyzing the mechanism between edges and nodes, writing a matrix represent the specific adjacent matrix in hypergraph as shown in Fig 7. Each module contains its corresponding features and transfers them to fixed 5-dimensions through a LSTM model as a pre-procession step. Therefore, each module will represent a 5-dimensions embedding vector equally which guarantees each module (node) has the same express space. After generating the initial representation of each module (node), those module representation vectors go through a ***k*** layer of graph convolutional process. The forward propagation is updating by the following algorithm:

$$X_m^{l+1} = \sigma\left(D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}} X_m^l \Theta\right), \qquad (16)$$

where $X_m^l = [v_{m0}^l, v_{m1}^l, v_{m2}^l, v_{m3}^l, v_{m4}^l]$, which represents the feature of 5 modules and following with the equation of (6)-(11) which generates the predicted embedding vector. In the experiment, m represents each sample in the dataset, so in the equation of (6) to (11), the trainable parameters like ***W*** and bias item ***b*** will be accumulated and updated by new input.

In X_m^l, the upper *l* represents the *l*-th feature of the dataset and the lower *m* represents the iteration step. The equation denotes the propagation from *l* to *l*+1 as simplify. The expected prediction represents the embedding vector of corresponding modules. This predictive average removal rate is a single value, so these 5 modules vector (5*5 matrix) need to concatenate into one single dimension (25*1) as a readout process. The graph representation V (25*1) can be aggregated to calculate a graph-level output (average removal rate). This paper utilizes fully connected layers with non-linear active functions (such as ReLu, Sigmoid) to predict the average removal rate. By defining the Mean square error as the loss function and the Adam algorithm as the weight optimization method and setting 75% of the dataset as the training dataset and the rest 25% as the testing dataset, this experiment shows that the MSE of the training dataset is 0.0037 and the MSE of testing dataset is 0.0044. Besides, experiment the different number of GCN layer to verify the validity of the proposed model structure.

TABLE II
THE MSE WITH DIFFERENT GCN LAYER NUMBER

GCN layer number	Training Dataset MSE	Testing Dataset MSE
1	0.0037	0.0044
2	0.0031	0.0031
3	0.002	0.0064
4	0.0044	0.0005
5	0.0041	$4.5491*10^{-5}$

As shown in the table II, both the decrease of MSE in training dataset and testing dataset accompany with the increasing GCN layer number, therefore the GCN layer can efficiently extract the high order interaction feature from the hypergraph neural network.

5. CONCLUSION

This paper proposes a framework to organize the modularized structural information of complex equipment as a hypergraph structure. A novel model of a hypergraph network that combines the hypergraph structure and graph convolutional layer is proposed. This model is designed to capture the time-domain feature in the hypergraph for prediction. It utilizes the convolutional process to the hypergraph for extracting high order feature which includes the interaction among components. Based on the proposed industrial hypergraph and the model, an experiment is conducted by using the CMP equipment as an example to predict the MRR. The results proved that the proposed approach has an expected result, and the efficiency of algorithm structure is proved by changing the depth of the model as shown in Table II. Hence, this hypergraph neural network takes complex modules correlation into representation learning, bridging the gap between single features and equipment modules.

The key contributions of this work are summarized as follows:

1) Provided a novel framework and methodology to modeling the equipment as the representative hypergraph. This directed hypergraph reflects the interaction between modules.

2) Proposed a hypergraph model. This model modularizes processes of the multi-time series data by grouping them into the

corresponding modules. It formulates complex and high order data correlation through its hypergraph structure.

3) Conducted an experiment with the hypergraph model in the CMP dataset to predict the MRR. Comparing with the different layer number in the model, it demonstrates the effectiveness of our proposed model.

Future research can focus on implementing this approach to other complex processes and problems.

ACKNOWLEDGEMENTS

This research work was partially supported by the grants from the National Natural Research Foundation of China (No. 52005424), and Research Committee of The Hong Kong Polytechnic University (G-UAHH), Hong Kong SAR, China.

REFERENCES

[1] C. J. Evans *et al.*, "Material removal mechanisms in lapping and polishing," *CIRP Ann. - Manuf. Technol.*, vol. 52, no. 2, pp. 611–633, 2003, doi: 10.1016/S0007-8506(07)60207-8.

[2] S. Hong, D. Han, J. Kwon, S. J. Kim, S. J. Lee, and K.-S. Jang, "Research paper Influence of abrasive morphology and size dispersity of Cu barrier metal slurry on removal rates and wafer surface quality in chemical mechanical planarization," *Microelectron. Eng.*, vol. 232, 2020, doi: 10.1016/j.mee.2020.111417.

[3] T. Fujita, "Evaluation of correlation between chemical modification state of pad and polishing rate in oxide chemical mechanical planarization," *Thin Solid Films*, vol. 709, no. June, p. 138233, 2020, doi: 10.1016/j.tsf.2020.138233.

[4] Q. Xu, L. Chen, J. Liu, and H. Cao, "A Wafer-Scale Material Removal Rate Model for Chemical Mechanical Planarization," *ECS J. Solid State Sci. Technol.*, vol. 9, no. 7, p. 074002, 2020, doi: 10.1149/2162-8777/abadea.

[5] C. Stuffle, L. V. Bengochea, Y. Sampurno, and A. Philipossian, "Ultra-Rapid Determination of Material Removal Rates Based Solely on Tribological Data in Chemical Mechanical Planarization," *ECS J. Solid State Sci. Technol.*, vol. 8, no. 5, pp. P3035–P3039, 2019, doi: 10.1149/2.0061905jss.

[6] H. Lee and H. Jeong, "A wafer-scale material removal rate profile model for copper chemical mechanical planarization," *Int. J. Mach. Tools Manuf.*, vol. 51, no. 5, pp. 395–403, 2011, doi: 10.1016/j.ijmachtools.2011.01.007.

[7] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009, doi: 10.1109/TNN.2008.2005605.

[8] J. Zhou *et al.*, "Graph Neural Networks : A Review of Methods and Applications," pp. 1–22.

[9] H. S. Lee, H. D. Jeong, and D. A. Dornfeld, "Semi-empirical material removal rate distribution model for SiO2 chemical mechanical polishing (CMP) processes," *Precis. Eng.*, vol. 37, no. 2, pp. 483–490, 2013, doi: 10.1016/j.precisioneng.2012.12.006.

[10] Z. Kong, A. Oztekin, O. F. Beyca, U. Phatak, S. T. S. Bukkapatnam, and R. Komanduri, "Process performance prediction for chemical mechanical planarization (CMP) by integration of nonlinear bayesian analysis and statistical modeling," *IEEE Trans. Semicond. Manuf.*, vol. 23, no. 2, pp. 316–327, 2010, doi: 10.1109/TSM.2010.2046110.

[11] Z. Li, D. Wu, and T. Yu, "Prediction of Material Removal Rate for Chemical Mechanical Planarization Using Decision Tree-Based Ensemble Learning," *J. Manuf. Sci. Eng. Trans. ASME*, vol. 141, no. 3, pp. 1–14, 2019, doi: 10.1115/1.4042051.

[12] P. Wang, R. X. Gao, and R. Yan, "A deep learning-based approach to material removal rate prediction in polishing," *CIRP Ann. - Manuf. Technol.*, vol. 66, no. 1, pp. 429–432, 2017, doi: 10.1016/j.cirp.2017.04.013.

[13] X. Jia, Y. Di, J. Feng, Q. Yang, H. Dai, and J. Lee, "Adaptive virtual metrology for semiconductor chemical mechanical planarization process using GMDH-type polynomial neural networks," *J. Process Control*, vol. 62, pp. 44–54, 2018, doi: 10.1016/j.jprocont.2017.12.004.

[14] W. C. Lih, S. T. S. Bukkapatnam, P. Rao, N. Chandrasekharan, and R. Komanduri, "Adaptive neuro-fuzzy inference system modeling of MRR and WIWNU in CMP process with sparse experimental data," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 71–82, 2008, doi: 10.1109/TASE.2007.911683.

[15] Y. Alsafi and V. Vyatkin, "Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing," *Robot. Comput. Integr. Manuf.*, vol. 26, no. 4, pp. 381–391, 2010, doi: 10.1016/j.rcim.2009.12.001.

[16] T. D. Hedberg, M. Bajaj, and J. A. Camelio, "Using Graphs to Link Data Across the Product Lifecycle for Enabling Smart Manufacturing Digital Threads," *J. Comput. Inf. Sci. Eng.*, vol. 20, no. 1, pp. 1–15, 2020, doi: 10.1115/1.4044921.

[17] C. Zhang, G. Zhou, Q. Lu, and F. Chang, "Graph-based knowledge reuse for supporting knowledge-driven decision-making in new product development," *Int. J. Prod. Res.*, vol. 55, no. 23, pp. 7187–7203, 2017, doi: 10.1080/00207543.2017.1351643.

[18] L. Tiacci, *Object-oriented event-graph modeling formalism to simulate manufacturing systems in the Industry 4.0 era*, vol. 99. Elsevier B.V., 2020.

[19] D. Zhou, J. Huang, and B. Sch, "Learning with Hypergraphs : Clustering , Classification , and Embedding," no. Figure 1.

[20] Y. Huang, Q. Liu, and D. Metaxas, "Video Object Segmentation by Hypergraph Cut Yuchi Huang," pp. 1738–1745, 2009.

[21] Y. Huang, Q. Liu, S. Zhang, and D. N. Metaxas, "Image Retrieval via Probabilistic Hypergraph Ranking Yuchi Huang," pp. 3376–3383, 2010.

[22] Y. Gao, M. Wang, D. Tao, and S. Member, "3-D Object Retrieval and Recognition With Hypergraph Analysis," vol. 21, no. 9, pp. 4290–4303, 2012.

[23] T. Hwang and J. Kocher, "Learning on Weighted Hypergraphs to Integrate Protein Interactions and Gene Expressions for Cancer Outcome Prediction," pp. 293–302, 2008, doi: 10.1109/ICDM.2008.37.

[24] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph Neural Networks. (arXiv:1809.09401v3 [cs.LG] UPDATED)," *arXiv Comput. Sci.*, [Online]. Available: http://arxiv.org/abs/1809.09401?utm_source=researcher_app&utm_medium=referral&utm_campaign=RESR_MRKT_Researcher_inbound.

[25] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN : Deep Learning on Spatio-Temporal Graphs," pp. 5308–5317.

[26] S. Ushakov and H. Zhang, "A Comprehensive Survey of Prognostics and Health Management Based on Deep Learning for Autonomous Ships," vol. 68, no. 2, pp. 720–740, 2019, doi: 10.1109/TR.2019.2907402.

[27] X. Huang, C. Zanni-merk, B. Cr, and B. Cr, "ScienceDirect Enhancing Deep Learning with Semantics : an application to Enhancing Deep Learning with Semantics : an application to manufacturing time series analysis manufacturing time series analysis," vol. 00, no. 2018, 2019, doi: 10.1016/j.procs.2019.09.198.

[28] J. Narwariya, P. Malhotra, V. TV, L. Vig, and G. Shroff, "Graph Neural Networks for Leveraging Industrial Equipment Structure: An application to Remaining Useful Life Estimation," 2018.

[29] N. Delhi *et al.*, "Predicting Remaining Useful Life using Time Series Embeddings based on Recurrent Neural Networks ∗," 2017, doi: 10.1145/nnnnnnn.nnnnnnn.

[30] G. S. B. B, P. Zhao, and X. Li, "Deep Convolutional Neural Network Based Regression Approach for Estimation," pp. 214–228, 2016, doi: 10.1007/978-3-319-32025-0.

[31] T. N. Kipf and M. Welling, "Semi-Supervised Classi- fication with Graph Convolutional Networks," pp. 1–14, 2017.