

GraSSNet: Graph Soft Sensing Neural Networks

Yu Huang^{1,2,†}, Chao Zhang^{1,3}, Jaswanth Yella^{1,4}, Sergei Petrov^{1,5}, Xiaoye Qian^{1,6}
¹Seagate Technology, ²Florida Atlantic University, ³University of Chicago, ⁴University of Cincinnati,
⁵Stanford University, ⁶Case Western Reserve University
 Email: {yu.1.huang, chao.1.zhang, jaswanth.k.yella, sergei.petrov, xiaoye.qian}@seagate.com

Yufei Tang², Xingquan Zhu²
 Florida Atlantic University, FL, USA
 Email: {tangy, xzhu3}@fau.edu

Sthitie Bom^{1,*}
 Seagate Technology, MN, USA
 Email: sthitie.e.bom@seagate.com

Abstract—In the era of big data, data-driven based classification has become an essential method in smart manufacturing to guide production and optimize inspection. The industrial data obtained in practice is usually time-series data collected by soft sensors, which are highly nonlinear, nonstationary, imbalanced, and noisy. Most existing soft-sensing machine learning models focus on capturing either intra-series temporal dependencies or pre-defined inter-series correlations, while ignoring the correlation between labels as each instance is associated with multiple labels simultaneously. In this paper, we propose a novel graph based soft-sensing neural network (GraSSNet) for multivariate time-series classification of noisy and highly-imbalanced soft-sensing data. The proposed GraSSNet is able to 1) capture the inter-series and intra-series dependencies jointly in the spectral domain; 2) exploit the label correlations by superimposing label graph that built from statistical co-occurrence information; 3) learn features with attention mechanism from both textual and numerical domain; and 4) leverage unlabeled data and mitigate data imbalance by semi-supervised learning. Comparative studies with other commonly used classifiers are carried out on Seagate soft sensing data, and the experimental results validate the competitive performance of our proposed method.

Index Terms—Soft Sensing, Machine Learning, Multi-Label Classification, Imbalanced Learning, Graph Neural Network

I. INTRODUCTION

Industry 4.0, which encompasses the Internet of Things (IoT) and smart manufacturing, marries physical production and operations with smart digital technology, machine learning, and big data to create a more holistic and better connected ecosystem for industries that focus on high-tech manufacturing [1]. Due to the increase in complexity and cost, the manufacturing industry, such as semiconductor manufacturing [2], is becoming more and more complicated. To improve the production efficiency and quality control, the direct, fast, and accurate measurement and analysis/inspection of *key quality indicators* (KQIs) are in rising demand. In response, soft-sensing models have been developed to estimate/predict KQIs expediently during the past decades, which is usually formulated as a mathematical model with *easy-to-measure* auxiliary variables as inputs and *hard-to-measure* key indicators as outputs [3]. While soft-sensing models are of the process monitoring and diagnosis tasks, this paper mainly focus on

diagnostic applications [4], i.e. multi-label multivariate time series classification problem [5]–[7].

To establish a soft-sensing diagnostic model, two major categories of methodologies are widely adopted, mechanism/knowledge-based and data-driven-based method [8]. The former requires expert knowledge (or a wealth of experience) of detailed and accurate mechanism of the manufacturing process, which is hard to meet (acquire) with the increasing complexity of the industrial processes. In contrast, data-driven-based method (esp., deep learning models) is ‘*winning*’ in the field of soft sensing technology. The improvements in data availability and computational scale have been the dominant driving force behind data-driven modeling.

With the rapid development of smart digital technology and the wide use of the distributed control systems [9], more and more complex and ever-evolving process data are generated and stored in huge amounts, as monitoring sensors are increasingly installed in factories to measure real-time process status (e.g., temperature, pressure, etc.). Such data have the attributes of high nonlinearity, high-dimension, imbalance, and noise. How to make full use of industrial big data to effectively improve diagnostic performance, as well as avoid complicated feature engineering and learn abstract representation automatically, have become a challenging problem in developing cost-effective and scalable methods.

Traditional data-driven soft-sensing models like the kernel principal component analysis [10], support vector machine [11] and artificial neural networks [12] have been introduced for fault classification in industrial processes. However, such models show limitations in handling multi-mode, high-dimensional, noisy, and imbalanced data. Recently, deep neural networks have achieved breakthrough results and exhibit stronger capabilities in learning and representation over traditional methods, such as stacked auto-encoder [13], [14] and convolutional neural networks [15]. Despite a proliferation of research that applies deep learning approaches to soft sensing, there are several aspects that need to be further investigated, considering the multi-label multivariate time series classification scenario.

The industrial data obtained in practice is usually multivariate time-series data collected by soft sensors. It is challenging

[†] Work performed while at Seagate. * Corresponding Author.

since soft sensing models need to consider both intra-series temporal correlations and inter-series correlations jointly. Deep learning models, such as long short-term memory [16] and temporal convolution networks [17], have achieved promising results in temporal modeling. However, most of them ignore modeling the correlations among multiple time-series. Recently, some novel works [18] tried to learn both correlations by stacking graph convolution neural networks (GCN) [19] to temporal modules, where GCN was designed to capture inter-series relationships explicitly based on pre-defined sensor topology.

Modeling the label dependencies is important in soft sensing since the collected data are usually associated with multiple labels. In physical world, some combinations of labels are almost impossible to appear, while some are coincident with high possibility. Many previous classifiers are essentially limited, ignoring the complex topology between labels. This vitalizes research in exploring the label correlations, including graph learning models [20], [21], recurrent neural networks-based model [22], and attention mechanisms [23]. Graph-based models have been proven to be more effective in modeling label correlation [21], [24]. However, to the best of our knowledge, the label correlation is mostly explored in image classification, under-explored in soft sensing.

In this paper, a novel graph based soft-sensing neural network (GraSSNet) model is proposed for complex industrial process inspection by classifying KQIs (labels) given multi-variate time series. This paper presents the first empirical study of wafer inspection challenge addressed by the competition that we organized in frame of the IEEE BigData 2021 Cup. in *Soft Sensing at Scale - Seagate*¹. We formulate the problem as multi-label classification, since diagnostic KQIs are not mutually exclusive. The main contributions of this paper are:

- 1) GraSSNet is proposed to capture the intra-series temporal patterns and inter-series sensor correlations jointly in the spectral domain, where spectral representations hold clearer patterns and can be classified more effectively. GraSSNet enables a data-driven construction of dependency graphs for different time series without pre-defined topologies.
- 2) To make full use of various types of data, the dependency between textual information and numerical time series data is modeled through an attention mechanism.
- 3) Graph attention networks are used to propagate information between multiple labels to explicitly model the label dependencies, where the label correlation matrix is defined based on their co-occurrence patterns.
- 4) Extreme negative-positive imbalance and high unlabeled rate – which are typical challenges in soft sensing – are explicitly addressed by training the model through a joint loss function.

II. RELATED WORK

Our research sits at the intersection of soft sensing, multi-label classification, and imbalance learning. Three mature fields, each with a long history and rich body of research. While we cannot do justice to all three, we highlight the most relevant works below.

A. Deep Learning in Soft Sensing

Soft sensors are widely constructed in factories to realize process monitoring, quality prediction, and other important industrial applications [3], [25]. Recently, improvements in big data and computational scale have driven a proliferation of research that apply deep learning approaches to soft sensors. Autoencoders are usually adopted to extract feature representations [4] and handling missing data issues [26], [27]. Convolutional neural networks (CNNs) are suited for processing grid data in capturing local dynamic characteristics [28] or processing signals in the frequency domain [29]–[31]. Recurrent neural networks (RNNs) and their variants LSTMs and GRUs based soft sensors were developed to estimate variables with strong temporal patterns [32], and to cope with strong nonlinearity and dynamics of the process [33]. With various machine learning based soft sensing model proposed for different aspects, there is still much to be done to better apply the advanced methods in the soft sensing domain, especially to meet the ever-changing demands in practical industrial processes.

B. Multi-Label Classification

Multi-label classification is a fundamental and practical task in machine learning, where the aim is to predict a set of labels related to a sample. In most multi-label tasks, labels are treated in isolation and converted into a set of binary classification problems to predict whether each label of interest presents or not. Deep CNNs [34]–[36], RNNs/LSTMs [37], [38], or hybrid models [39] are widely used and have achieved promising results. However, a key characteristic that distinguishes the multi-label from multi-class classification is the combinatorics of the output space [21]. Many researchers attempted to regularize the prediction space by capturing label dependencies. Notable success was reported by explicitly modeling label dependencies via graph model [21], [40]–[42] or word embedding based on knowledge priors [21], [43], while some work implicitly modeled the label correlations via attention mechanisms [23], [44].

C. Imbalanced Classification

Another key characteristic of multi-label classification is the inherent positive-negative imbalance [45]. Most samples may contain only a small fraction of the candidate labels, implying that the number of positive samples per category will be much lower than the number of negative samples. Some re-sampling methods [46] were proposed by only selecting a more balanced subset. However, such methods are not suitable for handling imbalanced multi-label classification, since each sample contains many labels and re-sampling cannot change

¹<https://github.com/Seagate/BigDataChallenge>

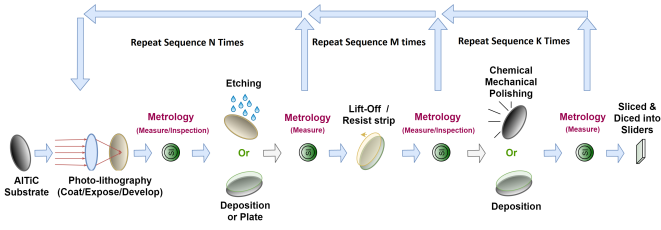


Fig. 1. Illustration of the wafer manufacturing process. Each wafer goes through multiple processing stages with corresponding meteorology where a few quality control measurements are performed. Figure from https://github.com/Seagate/softsensing_data.

the distribution of only a specific label [47]. Another common solution is to adopt a modified loss function [48] to train on all examples without sampling and without easy negatives overwhelming the loss and computed gradients. For example, focal loss [21], [49], [50] puts focus on hard samples while down-weighting easy samples, by decaying the loss as the label's confidence increases. More recently, asymmetric loss [47] focuses on hard negatives while maintaining the contribution of positive samples by decoupling the modulations of the positive and negative samples and assigning them with different exponential decay factors. It also shifts the probabilities of negative samples to completely discard very easy negatives.

III. PROBLEM DEFINITION

A. Soft Sensing at Scale - Seagate

The manufacturing process of wafers is complicated and time-consuming. As shown in Fig. 1, each wafer undergoes several permutations of the processing stages, including metal deposition, dielectric deposition, etching, electroplating, planarization, and lithograph [51]. Due to the multiple complicated processing stages, it is difficult to guarantee manufacturing stability at any time, which limits the quality control in actual industrial production. To improve the predictability of qualified product yield, a large sensor network is installed in the manufacturing line to monitor the wafer quality. At each processing stage, multiple critical sensor records are collected. The engineers at Seagate inspect these records and attest to the quality of the wafer based on some internal heuristic threshold values for each KQIs. However, the collected sensor data have the characteristics of high nonlinearity, dynamics, and noise, requiring a high labor force to handle. An efficient data-driven model is in demand to predict the inspection results (pass/fail of multiple binary indicators) based on the multivariate time-series sensor data. The above problem was presented in the big data challenge - *Soft Sensing at Scale - Seagate* - that we organized in frame of the IEEE BigData Cup 2021 [52]. The dataset released has 11 inspection KQIs (labels), with characteristics of high negative-positive rate, high unlabeled rate, and irregular time length. Statistic details of the dataset refer to Section VI-A.

B. Multi-Label Classification

Let $\mathbf{L} = \{l_1, \dots, l_N\}$ be a finite set of binary class labels and $\mathbb{X}$ denote an input space. Suppose every instance $x \in \mathbb{X}$, where $x \in \mathbb{R}^d$, is associated with a subset of labels $\bar{\mathbf{L}} \subset \mathbf{L}$, i.e., the set of relevant labels. The complement set of $\bar{\mathbf{L}}$ is called the irrelevant set. Therefore, $D = \{(x_1, \bar{\mathbf{L}}_1), (x_2, \bar{\mathbf{L}}_2), \dots, (x_n, \bar{\mathbf{L}}_n)\}$ is a finite set of training instances that are assumed to be randomly drawn from an unknown distribution. The objective is to train a multi-label classifier $f: \mathbb{X} \rightarrow 2^{\mathbf{L}}$ that best approximates the training data and generalizes well to the samples in the test data.

IV. PRELIMINARIES

Graph convolutional networks (GCNs) are a generalization of well-established convolutional neural networks to non-Euclidean graph-structured data. It can leverage graph topology to aggregate node information from the neighborhood in a convolutional fashion, following the widely-adopted GCN version proposed by [53], which is a spectral-based graph convolution design with spatial localization meaning. Assume we have a graph $\mathcal{G}$ with N nodes, whose topology is represented by an adjacency matrix $\mathcal{A} \in \mathbb{R}^{N \times N}$. By projecting the graph to an orthonormal space where the bases are constructed by eigenvectors of the normalized graph Laplacian, the corresponding normalized graph Laplacian is defined as,

$$\mathbf{L} = I_N - D^{-\frac{1}{2}} \mathcal{A} D^{\frac{1}{2}} = U \Lambda U^T \quad (1)$$

where $D \in \mathbb{R}^{N \times N}$ is the diagonal degree matrix with $D_{ii} = \sum_j \mathcal{A}_{ij}$, $I_N \in \mathbb{R}^{N \times N}$ is the identity matrix, and U and Λ are eigenvectors and diagonal matrix of eigenvalues corresponding to Laplacian matrix $\mathcal{L}$, respectively. The spectral convolution on the graph is,

$$\mathcal{G}_\theta \star x = U \mathcal{G}_\theta U^T x \quad (2)$$

where $x \in \mathbb{R}^N$ is a graph feature vector and $\mathcal{G}_\theta$ is the graph convolution kernel. The intuitive explanation is that a graph Fourier transform is first applied on graph features by $U^T x$, and then multiplied by the convolution kernel $\mathcal{G}_\theta$, and finally an inverse Fourier transform is performed by multiplying it with U . Therefore, the operators of Graph Fourier Transform (GFT) and Inverse Graph Fourier Transform (IGFT) are defined as,

$$\mathcal{GF}(x) = U^T x = \hat{x}; \quad \mathcal{GF}^{-1}(\hat{x}) = U \hat{x} \quad (3)$$

Next, we regard the convolution kernel as a polynomial function $\mathcal{G}_\theta(\Lambda)$ of the diagonal eigenvalue matrix Λ , so that the convolution becomes,

$$\begin{aligned} \mathcal{G}_\theta \star x &= U \mathcal{G}_\theta(\Lambda) U^T x \\ &= U \left(\sum_{k=0}^K \theta_k \Lambda^k \right) U^T x = \sum_{k=0}^K \theta_k \Lambda^k x \end{aligned} \quad (4)$$

where K is the chosen order of polynomial approximation, and θ are trainable parameters. To further improve the computational efficiency, we approximate $\mathcal{G}_\theta(\Lambda)$ by its Chebyshev polynomials and set the order of approximation $K = 1$. The Chebyshev polynomials are recursively defined as $T_k(x) =$

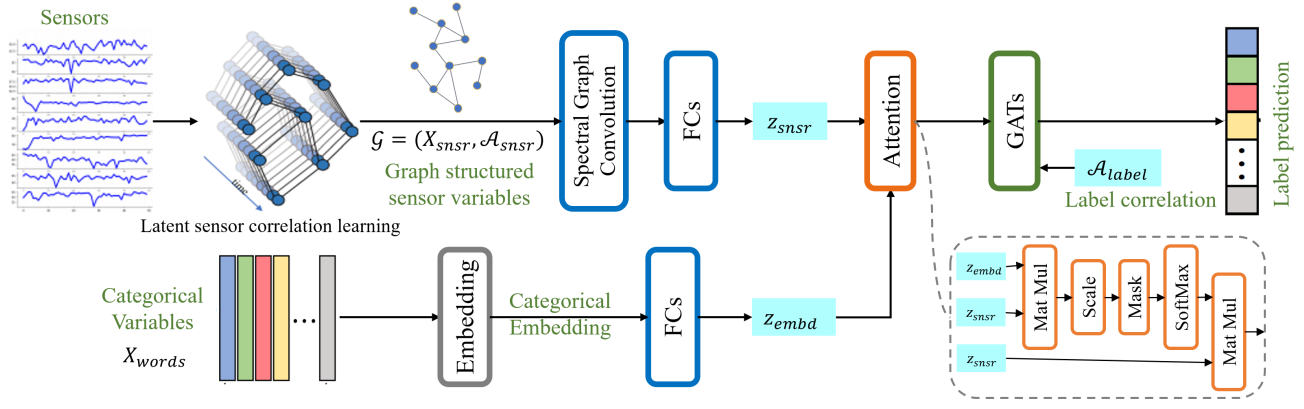


Fig. 2. Overall architecture of the proposed Soft-sensing Graph Neural Network (GraSSNet).



Fig. 3. Spectral Graph Convolution Module.

$2xT_{k-1}(x) - T_{k-2}(x)$, with $T_0(x) = 1$ and $T_1(x) = x$. To meet the requirement of Chebyshev polynomials, we normalize the eigenvalues as $\hat{\Lambda} = 2/(\lambda_{max}\Lambda - I_N)$ to make them lie within $[-1, 1]$. λ_{max} denotes the largest eigenvalue of $\mathbf{L}$, which is assumed to be 2. After a few derivation steps, the graph convolution becomes:

$$\begin{aligned} \mathcal{G}_\theta \star x &= \theta_0'x + \theta_1'(\mathbf{L} - I_N)x \\ &= \theta(I_N + D^{-\frac{1}{2}}\mathbf{A}D^{-\frac{1}{2}})x = \theta\tilde{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{D}^{-\frac{1}{2}}x \end{aligned} \quad (5)$$

with a single parameter $\theta = \theta_0' = -\theta_1'$, $\tilde{\mathbf{A}} = \mathbf{A} + I_N$ and $\tilde{D}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$. Though derived from the spectral domain, the graph convolution above is considered to have a clear meaning of spatial localization [54]. It is essentially equivalent to aggregating node representations from their direct neighborhood each time. Finally, the graph convolution cell can be defined as:

$$\mathbf{Y} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{D}^{-\frac{1}{2}}\mathbf{X}\Theta\right) \quad (6)$$

where $\mathbf{X}$ is the input, Θ is the trainable parameter matrix, and σ is the sigmoid activation function.

V. METHODOLOGY

A. Overall Framework

We propose a Graph based Soft-sensing Neural Network (GraSSNet) as a scalable solution for multivariate time-series classification in the soft sensing. The overall architecture of GraSSNet is illustrated in Fig. 2. It has two branches processing two types of data stream, i.e., numerical sensor records X_{snsr} and textual information X_{words} .

In the first branch, the multivariate time-series input X_{snsr} is first fed into a latent correlation layer to automatically infer the graph structure (i.e., the soft sensors network topology) and its associated weighted adjacent matrix $\mathbf{A}_{snsr}$. Next, the graph $G = (X_{snsr}; \mathbf{A}_{snsr})$ serves as input to the spectral

graph convolution module that is designed to model structural and temporal dependencies inside multivariate time-series jointly in the spectral domain (as visualized in Fig. 3). After spectral graph convolution module, feature representations on frequency basis are obtained by decomposing each individual time-series. Then, an output layer composed of fully-connected (FC) sub-layers is added to generate sensor feature representations with lower dimension. In the second branch, an embedding layer is used to encode lexical semantics, following with a FC sub-layers to learn a textual feature representations.

To prioritize and leverage the important distinctive features in sensor and textual feature representations, we introduce an attention mechanism to learn a feature fusion to boost performance. Finally, a graph attention network module is attached to capture the label correlations for multi-label classification and obtain the final predicted scores.

B. Latent Graph Learning Module

Graph neural network based approach requires a graph structure. It can be artificially constructed by human knowledge, such as using thresholded Gaussian kernel [55] to compute the pairwise road network distances between distributed sensors in traffic forecasting. However, sometimes we do not have a pre-defined graph structure as prior, such as in this paper, the sensor network topology is unknown. To tackle this problem, we leverage the self-attention mechanism to exploit the correlations between sensors, i.e. learn latent correlations between multivariate time-series automatically. In this way, the model emphasizes task-specific correlations in a data-driven fashion.

The multivariate time series X_{snsr} is first fed into a GRU layer, which calculates the hidden state corresponding to each time step t sequentially. Then, we use the last hidden state h as the representation of the entire time-series and calculate the weighted adjacent matrix $\mathbf{A}_{snsr}$ by the self-attention mechanism. An attention function can be described as mapping a query and a set of key-value pairs to an output [56]. The output is computed as a weighted sum of the values, where the

weight assigned to each value is computed by a compatibility function of the query with the corresponding key as follows,

$$\text{Query} = \mathbf{h}\mathcal{W}_{lg}^Q, \text{Key} = \mathbf{h}\mathcal{W}_{lg}^K$$

$$\mathcal{A}_{snr} = \text{Softmax}\left(\frac{\text{Query} \cdot \text{Key}^T}{\sqrt{d_K}}\right) \quad (7)$$

where Query and Key is calculated by linear projections with learnable weights $\mathcal{W}_{lg}^Q$ and $\mathcal{W}_{lg}^K$ in the attention mechanism, respectively; and d_K is the hidden dimension size of Key. The output matrix $\mathcal{A}_{snr} \in \mathbb{R}^{N \times N}$ is served as the adjacency weight matrix.

C. Spectral Graph Convolution Module

After obtaining the graph structured latent representation $\mathcal{G}(X_{snr}, \mathcal{A}_{snr})$ of the input multivariate time series, the graph $\mathcal{G}$ will be processed by a spectral graph convolution module, as shown in Fig. 3. This module is designed to model structural and temporal dependencies jointly in the spectral domain.

First, a Graph Fourier Transform (GFT) operator $\mathcal{GF}(\cdot)$ transforms the graph $\mathcal{G}$ into a spectral matrix representation on each individual channel X_i of input data, where the uni-variate time-series for each node becomes linearly independent. Then, the output of GFT is fed into the Discrete Fourier Transform (DFT), 1D convolution, Gated Linear Units (GLU), and Inverse Discrete Fourier Transform (IDFT) in order, aiming to decompose each individual sequence into frequency basis and learn feature representations on them. The DFT operator $\mathcal{DF}(\cdot)$ transforms each uni-variate time-series component into the frequency domain. In the frequency domain, the representation is fed into 1D convolution and GLU sub-layers to capture feature patterns in the frequency domain before transformed back to the time domain through IDFT $\mathcal{DF}^{-1}(\cdot)$. The process can be formulated as,

$$H_{snr} = \sum_i \mathcal{DF}^{-1}\left(\text{GLU}(\mathcal{DF}(\mathcal{GF}(X_{snr}^i)))\right)$$

$$= \sum_i \mathcal{DF}^{-1}\left(\text{GLU}(\theta_\tau^{re} \hat{X}_u^{re}, \theta_\tau^{im} \hat{X}_u^{im})\right) \quad (8)$$

$$= \sum_i \theta_\tau^{re} \hat{X}_u^{re} \odot \sigma\left(\theta_\tau^{im} \hat{X}_u^{im}\right)$$

where $\hat{X}_u^{re}$ and $\hat{X}_u^{im}$ are the real part and imaginary part of the output of DFT, which are processed by the same operators with different parameters θ_τ in parallel. θ_τ^{re} and θ_τ^{im} are the convolution kernels. $\odot$ is the Hadamard product and nonlinear sigmoid gate $\sigma(\cdot)$ determines how much information in the current input is closely related to the sequential pattern.

The output from the spectral graph convolution module is fed into fully-connected layers (FCs) to generate sensor features z_{snr} . The FCs composed of 1 layer normalization [57], 1 LeakyReLU activation layer, 1 dropout layer, and 2 stacked linear layers in order.

D. Leverage Textual Information

In soft sensing, except the ‘hard’ sensor types data (including, radar, multi-spectral, acoustic sensor array, etc), the ‘soft’ sensor inputs such as textual reports, and hybrid ‘hard/soft’ data such as human-annotated sensor data can be highly useful. It is worth categorizing and exploiting to get richer information to enhance the classifier. In this paper, the ‘soft’ sensor inputs refer to the textual information of the multi-stage manufacturing process (i.e., categorical variables), where the order of categorical variables is of importance. Textual information should be represented as a fixed-length vector without losing the semantics of the words [58], [59].

Similarly to other sequence transduction models, we use learnable embedding to convert the input tokens and output tokens to vectors of dimension d_{embd} . Then, we use a linear layer to map into a hidden textual information representation z_{embd} as,

$$z_{embd} = \text{Embedding}(X_{words}) * \mathcal{W}_{embd} + b_{embd} \quad (9)$$

where X_{words} is the input tokens of words, $\mathcal{W}_{embd}$ and b_{embd} are the weights and bias of the subsequent linear layer, respectively.

In classification problem, not all feature types are equally contributed to the classification task. In order to prioritize the important feature, as shown in Fig. 2, we introduce an attention mechanism to capture the dependencies of sensor features z_{snr} and textual feature z_{embd} . The output feature z_{att} is computed as a weighted sum of the z_{snr} , where the weight assigned to each dimension is computed by a compatibility function of the z_{embd} with the corresponding z_{snr} as follow,

$$z_{att} = \text{Attention}(z_{embd}, z_{snr}, z_{snr})$$

$$= \text{Softmax}\left(\frac{z_{embd} z_{snr}^T}{\sqrt{d}}\right) z_{snr} \quad (10)$$

where d is the dimension of z_{snr} , equals to the dimension of z_{embd} . We compute the dot products of the textual feature with all sensor features, divide each by $\sqrt{d}$, and apply a Softmax($\cdot$) function to obtain the weights on z_{snr} .

E. Label Correlation

In this paper, we use graph attention networks (GATs) [60] to model the inter dependencies between labels. GATs works by information propagation between nodes based on the correlation matrix $\mathcal{A}_{label}$, which is a flexible way to capture the topological structure in the label space. The correlation matrix is not provided in any standard multi-label time series classification datasets. Here, we construct a directed correlation matrix $\mathcal{A}_{label}$ via mining label’s co-occurrence patterns within the data. The label correlation dependency is modeled by the form of conditional probability, i.e., $P(l_j|l_i)$ which denotes the probability of occurrence of label l_j when label l_i appears. It is worth mentioning that $P(l_j|l_i)$ is not equal to $P(l_i|l_j)$. For example, when l_i appears in the sample, l_j will also occur with a high probability. However, in the condition of l_j appearing, l_i may not necessarily occur. In other words, the label correlation matrix is asymmetrical.

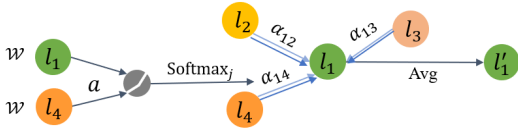


Fig. 4. An illustration of multi-head attention (with 2 heads) by node 1 on its 3 neighborhood.

To construct $\mathcal{A}_{label}$, firstly, we count the occurrence of label pairs in the training set and get the matrix $M \in \mathbb{R}^{C \times C}$, where C is the number of labels, and M_{ij} denotes the concurring times of l_i and l_j . Then, we can define the conditional probability matrix by $p_i = M_i/N_i$, where N_i denotes the occurrence times of l_i in the training set, and $p_{ij} = p(l_j|l_i)$ means the probability of label l_j when label l_i appears.

However, the co-occurrence patterns between one label and the other labels may exhibit a long-tail distribution, where some rare co-occurrences may be noise. Such a correlation matrix will over-fit the training data and thus hurt the generalization capacity. Specifically, a threshold τ is employed prior to filter noisy edges,

$$\mathcal{A}_{label}^{ij} = \begin{cases} 0 & p_{ij} < \tau \\ 1 & p_{ij} \geq \tau \end{cases} \quad (11)$$

where $\mathcal{A}_{label}$ is the binary correlation matrix.

The motivation of using GATs to model the inter dependencies between labels is by computing the score of each label l_i (node in graph) by attending over its co-occurrence label (neighbors) following a self-attention strategy, as shown in Fig. 4. we perform self-attention on the nodes as,

$$e_{ij} = a(\mathcal{W}l_i, \mathcal{W}l_j) \quad (12)$$

where $\mathcal{W}$ is a weight matrix; a is a shared attention mechanism $a: \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ for computing attention coefficients e . e_{ij} indicates the importance of node i to node j . We inject the graph structure into the mechanism by performing masked attention, i.e., only compute e_{ij} for nodes $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ is some neighborhood of node i in the graph.

To make coefficients easily comparable across different nodes, softmax function is applied across all choices of j . Thus, the normalized coefficients can be obtained by,

$$\alpha_{ij} = \text{Softmax}(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})} \quad (13)$$

Then, the normalized attention coefficients α are used to compute a linear combination of the labels scores corresponding to them, to serve as the final output score l' for every node,

$$l'_i = \sigma \left(\frac{1}{K} \sum_{k=1}^K \sum_{j \in \mathcal{N}_i} \alpha_{ij}^k \mathcal{W}^k l_j \right) \quad (14)$$

where K indicates the number of independent attention mechanisms, i.e. multi-head attention. α_{ij}^k are normalized attention coefficients computed by the k -th attention mechanism (a^k), and $\mathcal{W}^k$ is the corresponding input linear transformation's

weight matrix. We employ averaging, and delay applying the final nonlinearity activation σ .

F. Imbalanced Loss Function

In a typical multi-label setting, a sample may contain on average few positive labels, and many negative ones. This positive-negative imbalance dominates the optimization process, and can lead to under-emphasizing gradients from positive labels during training, resulting in poor accuracy. Here, we reduce the problem to a series of binary classification tasks. Given K labels, the base network outputs one logit per label, p_l^k . Each logit is independently activated by a sigmoid function. Let's denote y_l^k as the ground-truth for class k . The total supervised classification loss, $\mathcal{L}_l$, is obtained by aggregating a binary loss from K labels,

$$\begin{aligned} \mathcal{L}_l &= \sum_{k=1}^K -y_l^k * \mathcal{L}_{pos}^k - (1 - y_l^k) * \mathcal{L}_{neg}^k \\ \mathcal{L}_{pos}^k &= -(1 - p_l^k)^{\gamma_+} \log(p_l^k) \\ \mathcal{L}_{neg}^k &= -(p_l^k)^{\gamma_-} \log(1 - p_l^k) \end{aligned} \quad (15)$$

where y_l is the ground-truth label, and $\mathcal{L}_{pos}$ and $\mathcal{L}_{neg}$ are the positive and negative focal loss parts (following [49]), respectively. $p_l = \sigma(z)$ is the network's output probability and γ is the focusing parameter for inner trade-off. $\gamma_+ = \gamma_- = 0$ yields binary cross-entropy. By setting $\gamma > 0$, the contribution of easy negatives (having low probability, $p \ll 0.5$) can be down-weighted in the loss function, enabling to focus more on harder samples during training. Instead of using uniform γ , we decouple the focusing levels of the positive and negative samples by employing γ_+ and γ_- be the positive and negative focusing parameters, respectively.

However, we found that simple linear weighting is insufficient to tackle the negative-positive imbalance issue in our case. Instead, following the Asymmetric loss proposed in [47], we use an asymmetric focusing mechanism – probability shifting – to perform hard thresholding of very easy negative samples. Which means the negative samples will be fully discarded if their probability is very low. The asymmetric probability-shifted focal loss is defined as,

$$\mathcal{L}_{neg} = (\max(p - m, 0))^{\gamma_-} \log(1 - \max(p - m, 0)) \quad (16)$$

where $\max(p - m, 0)$ is the shifted probability, i.e., moving the loss function to the right by a factor m , where $\mathcal{L}_{neg} = 0$ if $p < m$.

Another major concern in multi-label classification is the high unlabeled rate. Semi-supervised learning provides an effective means of leveraging unlabeled data to improve a classifier's performance. This domain has witnessed rapid progress recently, at the cost of requiring more complexity in models [61]. Inspired by FixMatch proposed in [61], we introduce a semi-supervised loss term calculated on unlabeled samples by pseudo-labeling method, which uses the model's prediction as a 'label' to train against. Pseudo-labeling leverages the idea of using the model itself to obtain artificial labels for unlabeled data [62]. Specifically, this refers to leveraging "hard" labels

(i.e., the arg max of the model's output) and only retaining artificial labels whose largest class probability fall above a predefined threshold ς . Let p_u denotes the model's output of unlabeled sample, H denotes the cross-entropy between two probability distributions, $\mathbb{I}$ be the mask, then, the loss term of unlabeled samples is defined as follow,

$$\mathcal{L}_u = \sum_{k=1}^K \mathbb{I}(p_u^k > \varsigma) H(\hat{p}_u^k, p_u^k) \quad (17)$$

where $\hat{p}_u^k = \arg \max(p_u^k)$ and ς is the threshold. We assume that arg max applied to a probability distribution produces a valid "one-hot" probability distribution.

Finally, the overall loss function is defined as the sum of supervised loss $\mathcal{L}_l$ and unlabeled loss $\mathcal{L}_u$.

VI. EXPERIMENT

A. Seagate Soft-sensing Data

The data set is provided by the Seagate manufacturing factories in both Minnesota and Ireland, containing high dimensional time-series sensor data coming from different manufacturing machines. The textual information is the process-relevant categorical variables corresponding to the time-series data. The dataset is publicly available at https://github.com/Seagate/softsensing_data.

As shown in Fig. 1, an AlTiC wafer goes through multiple processing stages including polishing, deposition, lithography and etching. After each processing stage, the wafer is sent to metrology tools for quality control inspection, i.e., KQIs. Each metrology stage usually contains a few different measurements, and the same measurement may be performed in different stages. Given there are many-to-many mapping between processes and inspections in each stage, one sensor record are mapped to several KQIs. Each KQI contains a few numerical values to indicate the quality condition of the processed wafer, and a decision of pass/fail is made based on these numbers by engineers at Seagate. For the sake of simplicity, we only cover the pass/fail binary information for each KQI. So that each sample of time-series sensor data are mapped to several binary labels, resulting in a simplified multi-label classification problem. The statistics of the dataset is summarized in TABLE I. There are 11 KQIs (labels), and about 1.2% of them are positive (failed) samples. There are total 194k data samples for training, 34k samples for validation, and 27k samples for testing. The unlabeled rate of training dataset among each labels is list in TABLE I column 2. Zero padding is employed in data pre-processing to ensure each sample has fixed 2 time steps.

B. Baselines

- 1) Diagnose recurrent neural network (LSTM) [38] is the first study to empirically evaluate the ability of LSTMs to recognize patterns in multivariate time series of clinical measurements.
- 2) ML-GCN [21] is a multi-label classification model based on Graph Convolutional Network (GCN) that is desirable to model the label dependencies to improve the

TABLE I
IEEE BIG DATA CHALLENGE: SOFT SENSING AT SCALE - SEAGATE
DATASET STATISTICS

Labels (KQIs)	Unlabeled Rate	Train		Valid		Test	
		Neg	Pos	Neg	Pos	Neg	Pos
KQI-1	0.97	6020	272	1417	13	878	10
KQI-2	0.95	10288	33	1509	5	950	2
KQI-3	0.78	42989	200	7795	43	5414	48
KQI-4	0.94	11114	132	1594	23	1989	33
KQI-5	0.83	32794	428	4283	91	3567	49
KQI-6	0.67	64007	709	11833	68	9123	86
KQI-7	0.39	117332	1702	19663	482	16975	371
KQI-8	0.99	1748	443	196	39	975	8
KQI-9	0.88	22420	86	4225	6	2906	12
KQI-10	0.96	7874	48	1788	4	1151	5
KQI-11	0.81	35874	227	6231	36	5114	43

* Neg/Pos: the passed/failed samples of corresponding key indicator.

recognition performance. Here, we use two stacked convolution layers as feature extraction backbone and use categorical variables embedding as label representations. The label correlation matrix is set as described in Section V-E with $\tau = 0.4$.

C. Evaluation Metric

In order to evaluate our model comprehensively and for the convenience of comparison with other solutions, we report the average per-label, recall (L-R), area under receiver operating curve (L-AUC), the average overall recall (O-R), overall false alarm ratio (O-F), overall AUC (O-AUC) to estimate their effectiveness. For time series sample, the labels are predicted as positive if the confidences of them are greater than 0.5.

D. Implementation Details

All the baselines and proposed GraSSNet are trained on AWS p3.2x large instance with 16 GB NVIDIA Tesla V100 GPU. We implement the models based on PyTorch. Unless otherwise stated, we set $\tau = 0.4$ for the correlation matrix in Eq. (11), set $\varsigma = 0.95$ in Eq.(17); we adopt LeakyReLU [63] with the negative slope of 0.2 as the non-linear activation function. The dropout rate is 0.2. The dimension is 16 for embedding, 64 for z_{snst} and z_{emb} , which is chosen from a search space of [8, 16, 32, 64, 128] on the validation data. For network optimization, RMSProp [64] is used as the optimizer with $1e-4$ weight decay and $1e-3$ initial learning rate. The early stopping mechanism will be executed if the performance of the model on the validation dataset starts to degrade (with patience equals to 25 epochs).

VII. RESULTS AND DISCUSSION

Quantitative results are reported in TABLE II. We compare with state-of-the-art methods, including LSTM, ML-GCN multi-label classification tasks. Here, the L-AUC score for each label is used as the evaluation metric. Here we observe that GraSSNet model outperforms the baseline models in most of the labels prediction (highlighted in TABLE II), which shows the superiority of our proposed model. By comparing

TABLE II
AUC SCORE OF GRASSNET AND BASELINES ON TEST DATASET

Labels	GraSSNet	ML-GCN	LSTM
KQI-1	0.76 (± 0.027)	0.48(± 0.083)	0.61(± 0.007)
KQI-2	0.57(± 0.056)	0.60 (± 0.088)	0.43(± 0.056)
KQI-3	0.78 (± 0.074)	0.65(± 0.031)	0.48(± 0.037)
KQI-4	0.86 (± 0.001)	0.39(± 0.065)	0.49(± 0.006)
KQI-5	0.59(± 0.036)	0.63 (± 0.021)	0.44(± 0.024)
KQI-6	0.61(± 0.025)	0.83 (± 0.019)	0.53(± 0.028)
KQI-7	0.67 (± 0.003)	0.58(± 0.014)	0.51(± 0.035)
KQI-8	0.78 (± 0.057)	0.70(± 0.039)	0.38(± 0.024)
KQI-9	0.87 (± 0.029)	0.84(± 0.025)	0.63(± 0.009)
KQI-10	0.90 (± 0.032)	0.58(± 0.196)	0.46(± 0.009)
KQI-11	0.84 (± 0.045)	0.71(± 0.032)	0.62(± 0.056)

L-AUC scores, GraSSNet consistently outperforms LSTM. The major reason lies in that LSTM only takes temporal information into consideration and performs modeling in the time domain, while GraSSNet models the time-series data in the frequency domain and shows stable improvement over LSTM.

It is noteworthy that, both GraSSNet and ML-GCN that consider label correlation outperforms the LSTMs. It validates the effectiveness of using graphs to model the inter dependencies between labels to improve the classification performance. In both methods, a directed graph is built over labels representations where each node denotes a label, which is a flexible way to capture the topological structure in the label space. Furthermore, GraSSNet outperforms ML-GCN in 8 out of 11 KQIs classification. It shows the advantages of leveraging GFT to capture structural information in a graph combined with leveraging DFT to learn temporal patterns, when ML-GCN only use convolutional kernels to extract features.

VIII. ABLATION STUDY

In this section, we perform ablation studies from four different aspects, including the advantage of using textual information, effects of label correlation, effects of different loss functions in $\mathcal{L}_l$ for imbalanced multi-label classification, and effects of unlabeled data (with/without $\mathcal{L}_u$).

A. Textual Information - Categorical Variables

We illustrate that leveraging relevant textual data sources have the potential to improve multi-label classification performance in two ways, data distribution analysis and model performance. To visualize the data distribution, the Isomap [65] is applied, which is proposed for computing a quasi-isometric, low-dimensional embedding of a set of high-dimensional data points. As shown in Fig.5, in the embedded 2D observation space, although the categorical variables result in the fewest feature points, the classifier should not only depend on categorical variables since it only describes how a wafer goes through multiple processing stages. There is no causal relationship between the categorical variables and measurement outcome. As supplementary information, by leveraging

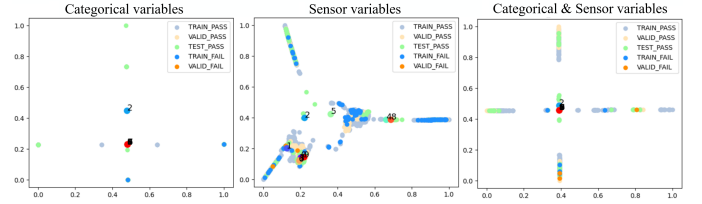


Fig. 5. Exemplarily data visualization by Isomap embedding method of label KQI-1 by categorical variables only, sensor variables only and both.

TABLE III
EFFECTIVENESS OF CATEGORICAL VARIABLES FOR GRASSNET

model	Cat vars	O-R	O-F	O-AUC
GraSSNet	w/	0.51 (± 0.043)	0.14 (± 0.009)	0.75 (± 0.016)
	w/o	0.31(± 0.129)	0.24(± 0.097)	0.56(± 0.054)

TABLE IV
EFFECTIVENESS OF LABEL CORRELATION FOR GRASSNET

model	Label Corr	O-R	O-F	O-AUC
GraSSNet	w/	0.51 (± 0.043)	0.14 (± 0.009)	0.75 (± 0.016)
	w/o	0.42(± 0.055)	0.16(± 0.030)	0.71(± 0.042)

categorical variables, the merged data is more ‘classifiable’ than sensor data only. To further quantify the effectiveness, we did the ablation study of proposed Soft-sensing GNN w/o categorical variables. As shown in TABLE III, we can clearly observe that GraSSNet with categorical variables are obviously better than that without categorical variables. Hybrid textual information can be highly useful in Soft sensing.

B. Label Correlation

A Naive way to address the multi-label classification problem is to treat the labels in isolation, which means convert the multi-label problem into a set of binary classification problems to predict whether a label presents or not. However, this way ignores the topology structure between labels. Especially, in our case, there’s some causal relationship between labels. To explore the effectiveness of using GCN to propagate information between multiple labels and consequently learn the inter-dependent relationship for each of the labels, we did the ablation study of proposed GraSSNet w/ or w/o label correlation. As shown in TABLE IV, it is apparent that our proposed GraSSNet with label correlation observes improvements upon the one without label correlation. It approves that capturing the correlations between labels and modeling these label correlations to improve the classifier’s performance are both important for multi-label classification.

C. Loss Function term $\mathcal{L}_l$ for Imbalance

A key characteristic of multi-label classification is the inherent positive-negative imbalance. Most samples contain only a small fraction of the possible labels, implying that the number of positive samples per category will be, on average, much

TABLE V
VARIOUS $\mathcal{L}_l$ FOR NEGATIVE-POSITIVE IMBALANCE (PER-LABEL)

Labels	Cross-entropy			Focal loss [49]			Asymmetric [47]		
	L-R	L-F	L-AUC	L-R	L-F	L-AUC	L-R	L-F	L-AUC
KPI-1	0.00	0.000	0.73	0.00	0.022	0.76	0.00	0.012	0.79
KPI-2	0.50	0.118	0.65	0.50	0.098	0.57	0.50	0.095	0.58
KPI-3	0.54	0.049	0.88	0.67	0.171	0.78	0.67	0.143	0.79
KPI-4	0.61	0.119	0.87	0.45	0.123	0.86	0.42	0.098	0.88
KPI-5	0.25	0.280	0.50	0.39	0.187	0.59	0.41	0.231	0.56
KPI-6	0.34	0.082	0.62	0.36	0.127	0.61	0.34	0.179	0.55
KPI-7	0.32	0.141	0.75	0.53	0.364	0.67	0.74	0.509	0.67
KPI-8	0.56	0.099	0.92	0.50	0.105	0.78	0.25	0.082	0.67
KPI-9	0.33	0.059	0.58	0.68	0.129	0.87	0.36	0.078	0.72
KPI-10	0.00	0.000	0.99	0.80	0.027	0.90	0.40	0.004	0.84
KPI-11	0.59	0.160	0.84	0.73	0.223	0.84	0.86	0.233	0.88
Avg	0.37	0.101	0.75	0.51	0.144	0.75	0.45	0.151	0.72

* Cross-entropy: $\gamma = 0$; Focal loss: $\gamma_+ = \gamma_- = 2$; Asymmetric loss: $\gamma_+ = 0, \gamma_- = 2$.

lower than that of negative samples. In our case, according to TABLE I, the average imbalance ratio of Neg/Pos = 80.02 for training data (valid:85.63, test:74.73), and the label *KQI-2*, *KQI-3* and *KQI-9* has top 3 highest high negative-positive imbalance. Another key characteristic of our case study is mislabeling, which could be caused by 2 possible reasons: 1) the dataset collected from both the US and Ireland factories, a wafer could be mislabeled when scrutinized its quality due to consensus conflict which may arise across global engineering teams. 2) the binary label indicating pass/fail is hard encoded based on an internal heuristic threshold value, there exists the possibility of inherent corruption associated with reliance on the threshold.

In this section, we explore how various state-of-the-art loss functions in the supervised $\mathcal{L}_l$ term affect the model performance. Those loss functions are designed for statistically handling the imbalance and mislabeling in multi-label classification problem. By setting $\gamma_+ = \gamma_- > 0$ in Eq.(15), we can get the format of Focal loss. Focal loss [49] is a common solution to deal with the imbalance in object detection. It puts focus on hard samples, while down-weighting easy samples by decaying the loss as the label's confidence increases. By setting $\gamma_+ = 0$ in Eq.(15) and $\gamma_- > 0$ in Eq.(16), we can get the format of Asymmetric loss. Asymmetric loss enables [47] us to dynamically down-weight and hard-threshold easy negative samples, while also discarding possibly mislabeled samples.

Here, we compare three different loss functions used in $\mathcal{L}_l$: Binary Cross-entropy, Focal loss and Asymmetric loss. As shown in TABLE V, the focal loss achieves the best overall recall and AUC while the cross-entropy loss achieves the lowest false positive rate. For cross-entropy loss, 6 out of 11 labels have the lowest false positive rates and highest AUC scores. Furthermore, focal loss and asymmetric loss significantly outperform cross-entropy on this case, demonstrating the effectiveness of γ in balancing between negative and positive samples. However, in terms of the recall, which is more important in industrial quality review, we recommend using focal loss training in our proposed GraSSNet.

TABLE VI
EFFECTIVENESS OF UNLABELED DATA FOR GRASSNET

model	$\mathcal{L}_u$	O-R	O-F	O-AUC
GraSSNet	w/	0.51 (± 0.043)	0.14 (± 0.009)	0.75 (± 0.016)
	w/o	0.41(± 0.051)	0.14(± 0.018)	0.71(± 0.042)

* $\mathcal{L}_l$ Focal loss: $\gamma_+ = \gamma_- = 2$.

D. Leveraging Unlabeled data

Data-driven models usually achieve their strong performance through supervised learning, which requires a labeled dataset. However, as shown in TABLE I, the unlabeled rate (Avg: 0.83) is extremely high in our case. It's worth leveraging unlabeled data to improve model performance. In this paper, we generate pseudo-labels using the model's predictions on unlabeled samples. The pseudo-label is only retained if the model assigns a high probability to one of the possible labels, which is realized by using the loss term $\mathcal{L}_u$, shown in Eq.(17). TABLE VI shows that, by adding $\mathcal{L}_u$, GraSSNet obtains substantially better performance in terms of O-R (24% increase) and O-AUC (5.6% increase). Although the O-F remains the same, the *std* is halved, indicating a more stable performance.

IX. CONCLUSION

This paper proposes a novel graph based soft-sensing neural network (GraSSNet) for multivariate time-series classification based on the case of wafer inspection challenge, where the data is noised, highly imbalanced, and unlabeled. In GraSSNet, a spectral graph convolution module is introduced to capture the 'classifiable' intra-series temporal patterns and inter-series sensor correlations jointly in the spectral domain through discrete Fourier transform and graph Fourier transform. Through attention mechanism, the model can leverage both textual information and numerical time series data. In the end, a graph attention network is attached to learn inter-dependent labels prior label representations. Furthermore, we introduce semi-supervised learning based on pseudo-labeling to mitigate the requirement for labeled data by providing a simplified means of leveraging unlabeled data. We also investigate the effectiveness of various loss functions in balancing contributions between negative and positive samples of multi-label classification. Both quantitative and qualitative results validated the advantages of our GraSSNet for soft sensor modeling in actual industrial processes.

ACKNOWLEDGMENT

This research is partially supported by U.S. National Science Foundation under Grant Nos. IIS-1763452, CNS-1828181, and IIS-2027339. We sincerely thank Seagate Technology for the support on this study, the Seagate Lyve Cloud team for providing the data infrastructure, and the Seagate Open Source Program Office for open sourcing the data sets and the code. Special thanks to the Seagate Data Analytics and Reporting Systems team for inspiring the discussions.

REFERENCES

- [1] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," in *2015 12th International conference on fuzzy systems and knowledge discovery (FSKD)*. IEEE, 2015, pp. 2147–2152.
- [2] S.-K. S. Fan, C.-Y. Hsu, C.-H. Jen, K.-L. Chen, and L.-T. Juan, "Defective wafer detection using a denoising autoencoder for semiconductor manufacturing processes," *Advanced Engineering Informatics*, vol. 46, p. 101166, 2020.
- [3] Q. Sun and Z. Ge, "A survey on deep learning for data-driven soft sensors," *IEEE Transactions on Industrial Informatics*, 2021.
- [4] Y. Huang, Y. Tang, and J. Vanzwieten, "Prognostics with variational autoencoder by generative adversarial learning," *IEEE Transactions on Industrial Electronics*, 2021.
- [5] C. Zhang, J. Yella, Y. Huang, X. Qian, S. Petrov, A. Rzhetsky, and S. Bom, "Soft sensing transformer: Hundreds of sensors are worth a single word," 2021.
- [6] X. Qian, C. Zhang, J. Yella, Y. Huang, S. Petrov, M.-C. Huang, and S. Bom, "Soft sensing model visualization: Fine-tuning neural network from what model learned," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021.
- [7] J. Yella, C. Zhang, Y. Huang, X. Qian, S. Petrov, A. Minai, and S. Bom, "Soft-sensing conformer: A curriculum learning-based convolutional transformer," in *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 2021.
- [8] V. Venkatasubramanian, R. Rengaswamy, and S. N. Kavuri, "A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies," *Computers & chemical engineering*, vol. 27, no. 3, pp. 313–326, 2003.
- [9] Z. Geng, Z. Chen, Q. Meng, and Y. Han, "Novel transformer based on gated convolutional neural network for dynamic soft sensor modeling of industrial processes," *IEEE Transactions on Industrial Informatics*, 2021.
- [10] J. Dai, N. Chen, X. Yuan, W. Gui, and L. Luo, "Temperature prediction for roller kiln based on hybrid first-principle model and data-driven mw-dlwkpcr model," *ISA transactions*, vol. 98, pp. 403–417, 2020.
- [11] Y. Wang, D. Wu, and X. Yuan, "A two-layer ensemble learning framework for data-driven soft sensor of the diesel attributes in an industrial hydrocracking process," *Journal of Chemometrics*, vol. 33, no. 12, p. e3185, 2019.
- [12] Y. Xuefeng, "Hybrid artificial neural network based on bp-plsr and its application in development of soft sensors," *Chemometrics and Intelligent Laboratory Systems*, vol. 103, no. 2, pp. 152–159, 2010.
- [13] X. Yuan, S. Qi, and Y. Wang, "Stacked enhanced auto-encoder for data-driven soft sensing of quality variable," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 7953–7961, 2020.
- [14] C. Zhang and S. Bom, "Auto-encoder based model for high-dimensional imbalanced industrial data," *arXiv preprint arXiv:2108.02083*, 2021.
- [15] Y. Lei, X. Chen, M. Min, and Y. Xie, "A semi-supervised laplacian extreme learning machine and feature fusion with cnn for industrial superheat identification," *Neurocomputing*, vol. 381, pp. 186–195, 2020.
- [16] X. Yuan, L. Li, Y. A. Shardt, Y. Wang, and C. Yang, "Deep learning with spatiotemporal attention-based lstm for industrial soft sensor model development," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 5, pp. 4404–4414, 2020.
- [17] B. H. D. Koh, C. L. P. Lim, H. Rahimi, W. L. Woo, and B. Gao, "Deep temporal convolution network for time series classification," *Sensors*, vol. 21, no. 2, p. 603, 2021.
- [18] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.
- [19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.
- [20] M. Shi, Y. Tang, and X. Zhu, "Mlne: Multi-label network embedding," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 9, pp. 3682–3695, 2019.
- [21] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5177–5186.
- [22] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2285–2294.
- [23] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 464–472.
- [24] M. Shi, Y. Tang, X. Zhu, and J. Liu, "Multi-label graph convolutional network representation learning," *IEEE Transactions on Big Data*, 2020.
- [25] X. Qian, H. Chen, H. Jiang, J. Green, H. Cheng, and M.-C. Huang, "Wearable computing with distributed deep learning hierarchy: a study of fall detection," *IEEE Sensors Journal*, vol. 20, no. 16, pp. 9408–9416, 2020.
- [26] Y. Huang, Y. Tang, J. VanZwieten, and J. Liu, "Reliable machine prognostic health management in the presence of missing data," *Concurrency and Computation: Practice and Experience*, p. e5762, 2020.
- [27] F. Guo, W. Bai, and B. Huang, "Output-relevant variational autoencoder for just-in-time soft sensor modeling with missing data," *Journal of Process Control*, vol. 92, pp. 90–97, 2020.
- [28] K. Wang, C. Shang, L. Liu, Y. Jiang, D. Huang, and F. Yang, "Dynamic soft sensor development based on convolutional neural networks," *Industrial & Engineering Chemistry Research*, vol. 58, no. 26, pp. 11 521–11 531, 2019.
- [29] X. Yuan, S. Qi, Y. A. Shardt, Y. Wang, C. Yang, and W. Gui, "Soft sensor model for dynamic processes based on multichannel convolutional neural network," *Chemometrics and Intelligent Laboratory Systems*, vol. 203, p. 104050, 2020.
- [30] J. Wei, L. Guo, X. Xu, and G. Yan, "Soft sensor modeling of mill level based on convolutional neural network," in *The 27th Chinese Control and Decision Conference (2015 CCDC)*. IEEE, 2015, pp. 4738–4743.
- [31] X. Qian, H. Cheng, D. Chen, Q. Liu, H. Chen, H. Jiang, and M.-C. Huang, "The smart insole: A pilot study of fall detection," in *EAI International Conference on Body Area Networks*. Springer, 2019, pp. 37–49.
- [32] X. Zhang and Z. Ge, "Automatic deep extraction of robust dynamic features for industrial big data modeling and soft sensor application," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4456–4467, 2019.
- [33] W. Ke, D. Huang, F. Yang, and Y. Jiang, "Soft sensor development and applications based on lstm in deep neural networks," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–6.
- [34] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [35] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Exploiting multi-channels deep convolutional neural networks for multivariate time series classification," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 96–112, 2016.
- [36] Z. Chen, Y. Liu, J. Zhu, Y. Zhang, R. Jin, X. He, J. Tao, and L. Chen, "Time-frequency deep metric learning for multivariate time series classification," *Neurocomputing*, 2021.
- [37] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fns for time series classification," *Neural Networks*, vol. 116, pp. 237–245, 2019.
- [38] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzell, "Learning to diagnose with lstm recurrent neural networks," *arXiv preprint arXiv:1511.03677*, 2015.
- [39] Z. Guo, P. Liu, J. Yang, and Y. Hu, "Multivariate time series classification based on mcnn-lstm network," in *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*, 2020, pp. 510–517.
- [40] X. Li, F. Zhao, and Y. Guo, "Multi-label image classification with a probabilistic label enhancement model," in *UAI*, vol. 1, no. 2, 2014, pp. 1–10.
- [41] M. Tan, Q. Shi, A. van den Hengel, C. Shen, J. Gao, F. Hu, and Z. Zhang, "Learning graph structure for multi-label image classification via clique generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4100–4109.
- [42] T. Durand, N. Mehrasa, and G. Mori, "Learning a deep convnet for multi-label classification with partial labels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 647–657.
- [43] Y. Wang, D. He, F. Li, X. Long, Z. Zhou, J. Ma, and S. Wen, "Multi-label classification with label graph superimposing," in *Proceedings of*

the AAAI Conference on Artificial Intelligence, vol. 34, no. 07, 2020, pp. 12 265–12 272.

- [44] R. You, Z. Guo, L. Cui, X. Long, Y. Bao, and S. Wen, “Cross-modality attention with semantic graph embedding for multi-label classification,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 709–12 716.
- [45] C. Zhang and S. Bom, “Auto-encoder based model for high-dimensional imbalanced industrial data,” 2021.
- [46] K. Oksuz, B. Cam, S. Kalkan, and E. Akbas, “Imbalance problems in object detection: A review. arxiv e-prints p,” *arXiv preprint arXiv:1909.00169*, 2019.
- [47] E. Ben-Baruch, T. Ridnik, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, “Asymmetric loss for multi-label classification,” *arXiv preprint arXiv:2009.14119*, 2020.
- [48] T. Wu, Q. Huang, Z. Liu, Y. Wang, and D. Lin, “Distribution-balanced loss for multi-label classification in long-tailed datasets,” in *European Conference on Computer Vision*. Springer, 2020, pp. 162–178.
- [49] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.
- [50] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, “Learning semantic-specific graph representation for multi-label image recognition,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 522–531.
- [51] M. Quirk and J. Serda, *Semiconductor manufacturing technology*. Prentice Hall Upper Saddle River, NJ, 2001, vol. 1.
- [52] S. Petrov, C. Zhang, J. Yella, Y. Huang, X. Qian, and S. Bom, “Ieee big-data 2021 cup: Soft sensing at scale,” *arXiv preprint arXiv:2109.03181*, 2021.
- [53] T. N. Kipf and M. Welling, “Variational graph auto-encoders,” *arXiv preprint arXiv:1611.07308*, 2016.
- [54] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, “Graph convolutional networks: a comprehensive review,” *Computational Social Networks*, vol. 6, no. 1, pp. 1–23, 2019.
- [55] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains,” *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.
- [56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [57] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” *arXiv preprint arXiv:1607.06450*, 2016.
- [58] M. Shi, J. Liu, D. Zhou, M. Tang, and B. Cao, “We-lda: a word embeddings augmented lda model for web services clustering,” in *2017 IEEE international conference on web services (icws)*. IEEE, 2017, pp. 9–16.
- [59] M. Shi, J. Liu, B. Cao, Y. Wen, and X. Zhang, “A prior knowledge based approach to improving accuracy of web services clustering,” in *2018 IEEE International Conference on Services Computing (SCC)*. IEEE, 2018, pp. 1–8.
- [60] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.
- [61] K. Sohn, D. Berthelot, C.-L. Li, Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, “Fixmatch: Simplifying semi-supervised learning with consistency and confidence,” *arXiv preprint arXiv:2001.07685*, 2020.
- [62] H. Scudder, “Probability of error of some adaptive pattern-recognition machines,” *IEEE Transactions on Information Theory*, vol. 11, no. 3, pp. 363–371, 1965.
- [63] A. L. Maas, A. Y. Hannun, A. Y. Ng *et al.*, “Rectifier nonlinearities improve neural network acoustic models,” in *Proc. icml*, vol. 30, no. 1. Citeseer, 2013, p. 3.
- [64] S. Ruder, “An overview of gradient descent optimization algorithms,” *arXiv preprint arXiv:1609.04747*, 2016.
- [65] M. Balasubramanian, E. L. Schwartz, J. B. Tenenbaum, V. de Silva, and J. C. Langford, “The isomap algorithm and topological stability,” *Science*, vol. 295, no. 5552, pp. 7–7, 2002.