A Conditional Recurrent Autoencoder for Anomaly Detection in Overhead Hoist Transport Systems According to their Operational State

Jiyoon Myung and Young Jae Jang

Abstract— **Overhead hoist transport (OHT) systems are the primary systems used for the automated transportation of parts and materials in contemporary semiconductor fabrication facilities (FABs). OHT systems must be continuously monitored for the presence of abnormalities, so that these can be rapidly repaired to maintain system reliability. Since the OHT system is dynamic, it has various operational states such as driving and loading/unloading. Given that the operation of an OHT system depends on its state, this paper presents a framework for detecting anomalies in an OHT system according to state information. In particular, we describe a novel conditional recurrent autoencoder (CRAE) that can leverage its conditional input structure to process sensor data according to an OHT system's state. The performance of our CRAE-based method was verified with data collected from an OHT system test-bed, which showed that it effectively detected various state-dependent anomalies in this model OHT system.**

Keywords: OHT, anomaly detection, condition-based maintenance, autoencoder.

I. Introduction

The rapid and reliable transport of semiconductor wafers in contemporary semiconductor fabrication facilities (FABs) is underpinned by the interconnectedness of semiconductor manufacturing processes, which is achieved by the use of an overhead hoist transport (OHT) system that consists of OHT vehicles and a rail track system (hereinafter denoted "tracks"). OHT vehicles transport wafer-loaded front-opening unified pods (FOUPs) along the tracks from one machine to the next in a FAB, and load or unload their FOUP into a station or side-track buffer (STB). A station is the loading or unloading port for FOUP cargo, while the STB is a temporary storage place for a FOUP when the port is full. At a station, a component of an OHT vehicle called a hoist moves up and down to perform tasks, whereas, at the STBs, a component called a slide moves left and right to perform tasks.

The failure of an OHT system decreases the production rate of an entire manufacturing system, and therefore the condition of an OHT system must be continuously monitored to detect problems and thus enable their rapid resolution. Currently, OHT systems are typically maintained via a preventive approach, which is based on a checking system performance after predefined periods of operation rather than on a continuous, as-needed basis. In [1], an artificial neural network-based algorithm was developed to determine the real conditions of an OHT system using a measuring platform that captured three-dimensional data of power supply cables on the system's tracks. In [2], researchers presented a method to measure the current condition of an OHT vehicle, which collects the sensor data of the vehicle and then automatically detects and reports its abnormalities. In [2], a recurrent neural network-based autoencoder (RAE) was used to train the normal behavior of OHT vehicles. This generated a model that detected the current condition as abnormal if the online reconstruction error (RE) was above a given threshold. The results demonstrated that the RAE effectively processed the temporal sensor data of the OHT vehicles.

Figure 1. OHT vehicle used in semiconductor factories.

However, these abovementioned approaches did not consider the heterogeneous characteristics of an OHT system, which are a result of its dynamicity: the behavior of an OHT system depends on the current tasks of OHT vehicles. For example, traveling vehicles are propelled by a drive motor, while vehicles loading or unloading their FOUP are propelled by a hoist motor or a slide motor. In a given driving state, the motion or vibration of an OHT vehicle varies with the shape of the track (which may be straight, a curve, or S-shaped). In addition, the loading/unloading state can also be a station loading/unloading state or an STB loading/unloading state, according to the location of the OHT vehicle and the motor involved. Furthermore, an OHT vehicle-mounted FOUP affects the torque of the motors and thus the operational states of the system.

The state of an OHT vehicle dictates its behavior. This means that an OHT vehicle's state must be considered when examining an OHT system for anomalies, as without this consideration, anomalous operations in one state cannot be distinguished from normal operations in another state. For example, OHT vehicle vibrations are greater when a vehicle is traveling on a curved track than when it is traveling on a straight track. Thus, when such vibrations increase when an OHT vehicle is traveling on a track, it is difficult to determine whether this increase is normal (as the vehicle is traveling on a curved track) or abnormal (as the vehicle is traveling on a straight track) unless the shape of the track is known.

This research is supported by supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2021R1A2C3008172).

Jiyoon Myung is with Samsung SDS, South Korea

Young Jae Jang is with Department of Industrial & Systems Engineering, KAIST, South Korea.

An OHT management system (OMS) often misrecognizes an OHT vehicle's location when the vehicle fails to read the location indicator in the system. This common abnormality of an OHT system causes errors in vehicle operations. For example, if the OMS incorrectly recognizes a curved-driving OHT vehicle to be on a straight track, it will not instruct a fast-traveling vehicle to slow down at the speed limit of the curved track, and thus the vehicle may derail. These cases of false state recognition are called state anomalies in this paper, and as state anomalies can lead to serious system failures, they must be detected as soon as they occur. This must be done by determining whether if the recognized state is consistent with the current sensor data.

We address this need by developing a conditional recurrent autoencoder (CRAE) model for detecting anomalies in an OHT system by examining the system state. Our contributions are as follows. (1) We describe the need to consider system states for detecting anomalies in a dynamic system with various operating states. (2) We established a framework to identify the current state of an OHT vehicle from information managed by the OMS. (3) We develop a CRAE to process the recognized state together with the sensor data. (4) To validate the need to consider the state for anomaly detection, we designed several state-dependent anomalies that can be collected from an actual OHT test-bed. (5) We verified the effectiveness of the conditional structure of the CRAE for detecting OHT system anomalies using system state information.

II. Literature Review

A. Anomaly Detection in Industrial Systems

Anomaly detection has been widely applied to various industrial systems to prevent economic losses. It is also referred to as system fault detection or system diagnosis [3]. The data are generally in the form of sensor data, which are continuously monitored. Moreover, these are usually high-dimensional data because they are collected over multiple channels to represent the system condition. Deep learning-based models have been widely used in recent studies to manage these high-dimensional data. However, these deep learning techniques have been mainly applied to simple machinery or industrial systems without multiple operational states, such as wind turbines [4, 5], rotating mechanical parts [6–10], and other systems [11–14]. To apply anomaly detection in heterogeneous systems such as an OHT system, a structure that considers a system's state is required.

B. Deep Learning Models with a Conditional Structure

Some recent studies have shown that the conditional structure of a deep learning model can process input data according to additional contextual information. In [15], a conditional generative adversarial network (CGAN) was designed to generate new data conditioning based on additional contextual information such as class labels or data from other modalities. This CGAN performs conditioning by using the additional contextual information as an additional input layer in both the discriminator and generator. Several studies have used this model to generate additional abnormal data to address imbalance problems related to anomaly detection [16–18].

In [19], a conditional variational autoencoder (CVAE), which incorporates a conditional structure into an autoencoder (AE), was proposed. Similar to CGAN, CVAE can be implemented by feeding the additional contextual information into the encoder and decoder of the AE. Although the original CVAE was proposed to generate data, some studies have used CVAEs to detect anomalies in different contexts, given the effectiveness of AEs for anomaly detection. In [20], the authors used a CVAE to predict intrusion type for intrusion detection by integrating intrusion labels within the input layer of the decoder. Moreover, [21] detected anomalies of a complex muon solenoid trigger system by conditioning the trigger rate to consider that the second stage of the system was seeded by the trigger rate. In [22], a novel conditional convolutional AE (CCAE), which was based on the conditional structure of a CVAE, was developed for monitoring wind turbine blade breakages. In the CCAE, the conditional structure is applied to the AE with convolutional layers, and temporal sequence information is fed to the inputs of the encoder and decoder. As a result, the CCAE can detect breakages of the wind turbines according to the time of operation.

Thus, as a conditional structure effectively processes data according to the additional contextual information, we applied it to the RAE, using the state information as the context vector. This CRAE model considers the operational state of an OHT system and can detect anomalies in the system by processing state information together with sensor data. We collected some actual anomalies of an OHT system to validate the performance of our model. Consequently, the model was able to detect various state-dependent anomalies.

III. Methodology

This section presents a CRAE-based monitoring method for detecting anomalies in an OHT system having various operational states. The overall framework is summarized in Fig. 2. First, the Internet of Things (IoT) sensor data of OHT vehicles are collected, and then used as the input of the anomaly detection model. We develop a method to identify the current operational state of a system from the OMS. The CRAE is trained to reconstruct the sensor data by being conditioned on state information. In the offline training phase, the CRAE stores the REs of the normal data according to their corresponding state. In the online monitoring phase after training, the CRAE reports an anomaly if the current RE exceeds the normal range constructed in the training phase.

A. Identification of Current Operational State

To reflect the operational state in the detection of anomalies in an OHT system, the current state of the system must be identified. First, driving and loading/unloading states can be identified by the OMS from the activity of motor controllers. Second, assuming that the map information is known, the type of track (straight/curve) at the current location can be determined by examining the last recognized location sign. This information can be used to distinguish between station loading/unloading and STB loading/unloading. Third, by examining the hoist motor controller data, the OMS can determine whether a vehicle is currently carrying an FOUP. Therefore, in this paper, the system state is classified a straight-track driving, curved-track driving, slide operation (STB loading/ unloading), or hoist operation (station loading/unloading) state, and each state is further divided into an FOUP-equipped state or a non-FOUP-equipped state. Therefore, there are a total of eight states.

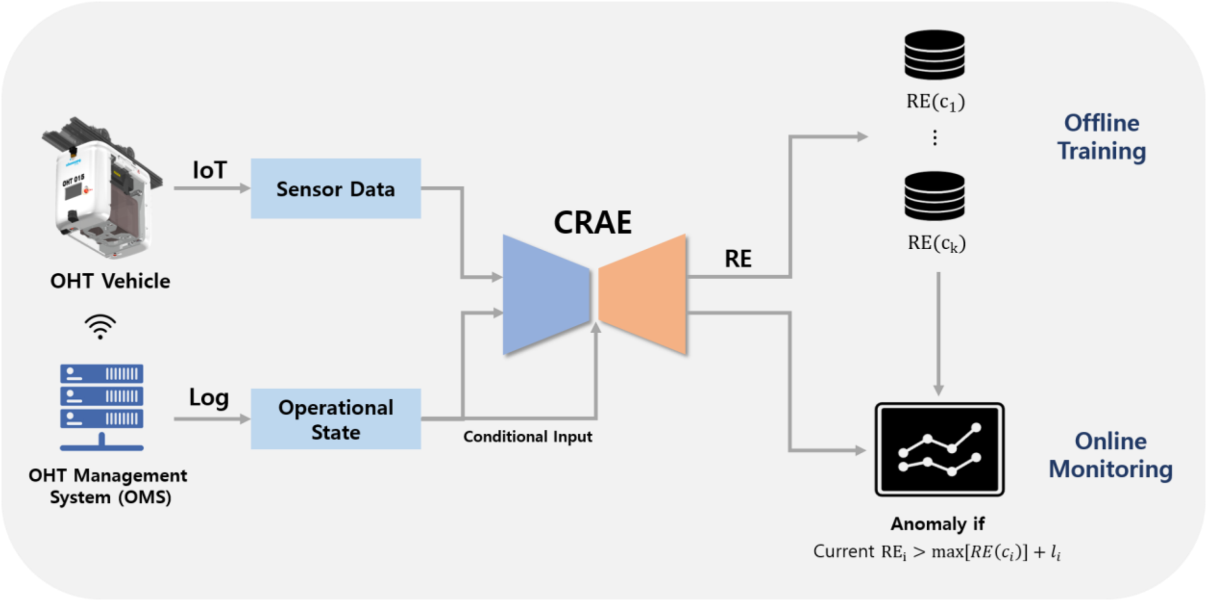


Figure 2. CRAE-based monitoring framework for detecting anomalies in OHT systems.

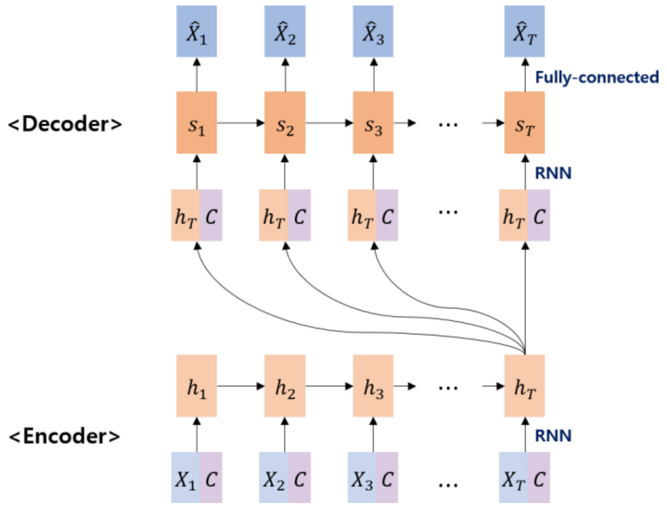


Figure 3. CRAE architecture.

B. Conditional Recurrent Autoencoder

We develop a CRAE, which is based on RAE architecture, to detect anomalies according to an OHT system's operational state. The overall architecture of the CRAE is presented in Fig. 3.

Consider an aggregated observation generated by the sliding window (SW) technique,

$$\mathbf{X} = [\mathrm{X}_1, \mathrm{X}_2, \dots, \mathrm{X}_t, \dots, X_T] \in \mathbb{R}^{n\times T} \tag{1}$$

, where T is the SW length, n is the number of sensor channels, and

$$\mathrm{X}_t = (\mathrm{x}_t^1; x_t^2; \dots; x_t^n) \in \mathbb{R}^{n\times 1} \tag{2}$$

is the observation of sensors at the singular timestamp t. As shown in the figure, a context vector C is also fed as the inputs of the encoder and decoder of the CRAE. The context vector is a one-hot vector of the length of the number of states, and corresponds to the current operational state. Each recurrent neural network (RNN) cell in the encoder at the time stamp t takes the concatenated input

$$\mathrm{Y}_\mathrm{t} = [\mathrm{X}_\mathrm{t}; \mathrm{C}] \in \mathbb{R}^{(n+k)\times 1} \tag{3}$$

, where k is the length of the context vector. Similarly, each cell in the decoder takes the input

$$\mathrm{I}_\mathrm{T} = [\mathrm{h}_\mathrm{T}; \mathrm{C}] \in \mathbb{R}^{(l+k)\times 1} \tag{4}$$

, where l is the length of the hidden output h_T. Therefore, the encoder and decoder of the CRAE with RNN cells can be illustrated as follows:

$$\mathrm{h}_\mathrm{t} = f(W_{enc}Y_t + U_{enc}h_{t-1} + b_{enc}) \tag{5}$$

$$\mathrm{s}_\mathrm{t} = f(W_{dec}I_T + U_{enc}s_{t-1} + b_{dec}) \tag{6}$$

, where $\mathrm{W}_\mathrm{enc} \in \mathbb{R}^{l\times(n+k)}$, $\mathrm{U}_\mathrm{enc} \in \mathbb{R}^{l\times l}$ and $\mathrm{b}_\mathrm{enc} \in \mathbb{R}^{l\times 1}$ represent the weight and bias of each RNN cell in the encoder, and $\mathrm{W}_\mathrm{dec} \in \mathbb{R}^{l\times(l+k)}$, $\mathrm{U}_\mathrm{dec} \in \mathbb{R}^{l\times l}$ and $\mathrm{b}_\mathrm{dec} \in \mathbb{R}^{l\times 1}$ are the weight and bias of the decoder. To realize the conditional structure, CRAE reconstructs the original sensor input X_t only, without state information C:

$$\hat{X}_t = W_{out}s_t + b_{out} \tag{7}$$

, where $\mathrm{W}_\mathrm{out} \in \mathbb{R}^{n\times l}$ and $\mathrm{b}_\mathrm{out} \in \mathbb{R}^{n\times 1}$ are the weight and bias of the output layer. Therefore, the RE of CRAE is expressed as that of the RAE:

$$\mathrm{L}(\mathrm{W}, \mathrm{b}, \mathrm{U}) = \frac{1}{\mathrm{T}}\sum_{t=1}^{T}\left\|X_t - \hat{X}_t\right\|^2 \tag{8}$$

With this architecture, the CRAE can process sensor data conditioning based on state information. Thus, the CRAE can detect anomalies in an OHT system by considering the sensor data with reference to the current operational state of the system.

C. Anomaly Detection Based on CRAE

As discussed above, the RE generated by the CRAE can be considered the anomaly score of an observation. Because the range of REs varies with operational state owing to the dynamic nature of OHT systems, we develop various thresholds for anomaly detection in different states. During the training phase, the REs of the training data are stored in terms of their states. Because all training data are considered normal, we assume that the errors obtained at this phase form a range of normal REs. In the detection phase, our CRAE model regards an observation as anomalous if its RE exceeds this range. Specifically, an observation is considered anomalous if its RE is greater than the threshold for the corresponding state $\mathrm{i} \in 1, \ldots, \mathrm{k}$:

$$\mathrm{Th_i} = \max RE(c_i) + l_i \tag{9}$$

, where $\mathrm{l_i}$ is a constant for sensitivity control for the ith state. The level of sensitivity control may vary depending on the FAB or situation.

IV. EXPERIMENT AND RESULTS

This section introduces an experiment we performed to evaluate our framework for anomaly detection in an OHT system. We adopted the RAE used in [2] as the main benchmark model to verify the ability of the CRAE to consider the state of a system.

A. Laboratory Environment

The experiment was conducted in a laboratory model of an actual semiconductor FAB. It contained a 50 m rail track, four OHT vehicles, three stations, and 14 STBs. The track consisted of straight and arced section, which means that the OHT vehicles could exist in one of two states: a straight-track driving state and a curved-track driving state. Fig. 4 illustrates the laboratory environment in detail. We embedded an IoT platform in the OHT vehicles to collect relevant sensor data.

B. Data Description

The collected sensor data consisted of 12 channels: vibrations in the x-, y-, and z-directions (X root-mean-square (RMS), Y RMS, and Z RMS, respectively); yaw (Yaw); and the speed and torque of the front wheels (Front SP, Front TQ), rear wheels (Rear SP, Rear TQ), hoist (Hoist SP, Hoist TQ), and slide (Slide SP, Slide TQ).

We first collected training data to train our model. These training data were normal data (for OHT vehicles driving normally), as the CRAE model was used in a semi-supervised manner. Data corresponding to the static state of a vehicle were excluded because these data were identical in normal and abnormal conditions. Because each sensor channel had a different range, it was normalized from 0 to 1 according to the minimum and maximum values of the training data. Fig. 5 shows the normalized samples for various operating states under normal conditions.

Because Yaw, Hoist SP, and Slide SP had symmetric values for 0, their values of approximately 0.5 in the normalized sample indicated a static state. The closer the value was to 0 or 1, the greater the absolute value was in a different direction. For example, a Yaw value close to 1 indicated a left turn, whereas a value close to 0 indicated a right turn. Similarly, a Hoist SP or Slide SP value close to 1 indicated an extension motion, whereas a value close to 0 indicated a retraction motion.

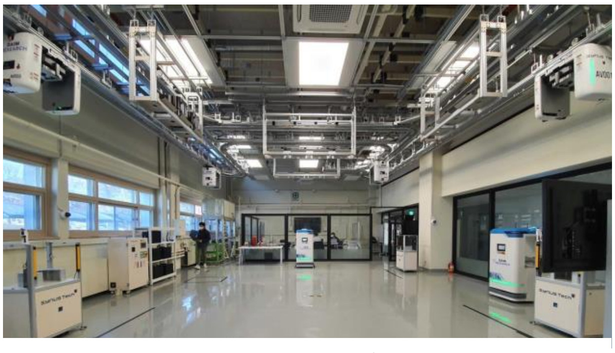
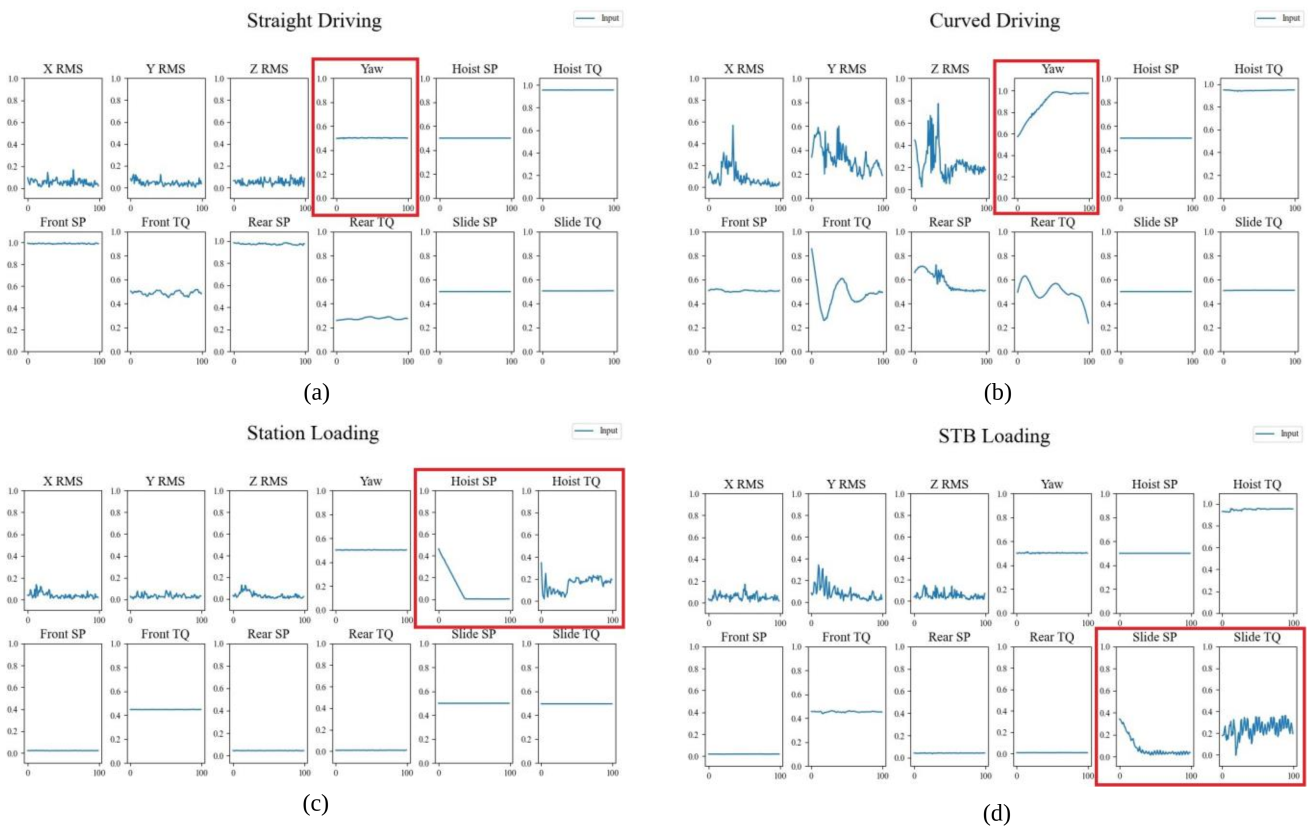


Figure 2. Laboratory environment for the OHT system.

Because the behaviors of the OHT vehicles were state-dependent, the sensor data were also state-dependent. For example, the Yaw value was close to 0.5 when an OHT vehicle traveled on straight tracks (Fig. 5a), whereas it approached 1 or 0 as an OHT vehicle traveled on curved tracks (Fig. 5b). In addition, when an OHT vehicle loaded/unloaded a FOUP at a station, its Front SP and Rear SP values remained almost equal to 0, but its Hoist SP and Hoist TQ values deviated from the static state (Fig. 5c). Similarly, its Slide SP and the Slide TQ values deviated from 0.5 in the STB loading/unloading state (Fig. 5d).

In addition, we collected test data to evaluate the anomaly detection performance of our model and a benchmark model. Unlike the training data, the test data consisted of both normal and abnormal data, and the purpose of the models was to distinguish these two data. The normal test data were collected in the same way as the training data. We collected abnormal data of several state-dependent anomalies to evaluate the ability of the model to consider the following states: Obstacles on a Straight Track, Increased Hoist Torque without an FOUP, Driving on a Straight Track + a Curved State, Driving on a Curved Track + a Straight State, and STB Operation + a Straight State.

First, the Obstacles on a Straight Track scenario indicated the data of OHT vehicles driving on straight tracks with obstacles. To reproduce this scenario, we added thin pieces of paper onto the track; the wheels of the OHT vehicles rattled as they traveled over this paper, resulting an increase in the vehicle vibration. Our model needed to be able to distinguish this scenario from normal curved-track driving because the vibrations occurring during curved-track driving are greater than those occurring during straight-track driving. Next, the Increased Hoist Torque without an FOUP scenario represents an increase in hoist torque occurring when the hoist was not holding an FOUP. This is because an increase in the hoist torque when the hoist moves without an FOUP indicates a defect in the hoist motor. This scenario cannot be distinguished from normal FOUP lifting without knowledge on the absence of an FOUP. Fig. 6 shows the normalized abnormal samples for each scenario. In the Obstacles on a Straight Track scenario (Fig. 6a), the RMS values related to vibration were greater than those of normal driving. Furthermore, in the Increased Hoist Torque without an FOUP scenario, the Hoist TQ value was higher than that under normal hoist operation (Fig. 6b).



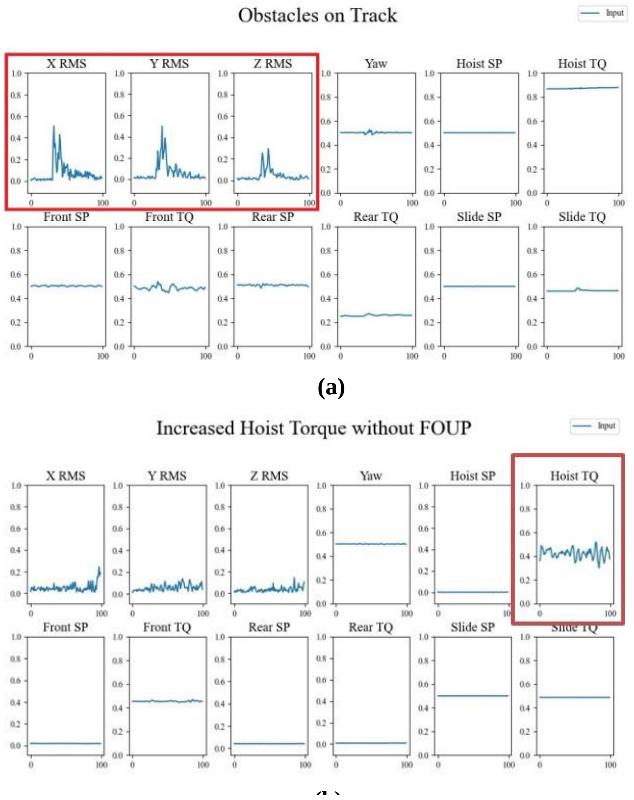
(a)

(b)

(c)

(d)

Figure 3. Normalized normal samples for different states with each relevant sensor channel indicated: (a) a straight-track driving state; (b) a curved-track driving state; (c) a station loading/unloading state; and (d) an STB loading/unloading state



(a)

Figure 6. Normalized abnormal samples for each scenario with representative sensor channels indicated: (a) obstacles on a straight track; and (b) increased hoist torque without an FOUP.

The last three scenarios represent state anomalies indicating cases where an OMS misrecognize an OHT vehicle's state. The first example of a state anomaly is the case of straight-track driving being recognized as curved-track driving (Straight Driving + a Curved State). We reproduced this scenario by manually combining the sensor data of normal straight-track driving with the state label of curved-track driving. The second example is the opposite case; that is, the sensor data of normal curved-track driving being recognized as straight-track driving (Curved Driving + a Straight State). These two cases often occur in actual FABs when an OHT vehicle fails to read the location sign of a control point on the track. The last example is a case of slide operation with a label of straight-track driving (Slide Operation + a Straight State). Although this scenario does not occur in actual FABs, we included this example to validate the performance of the models with a variety of examples. In these three cases, the sensor data are identical to normal data if system state information is not considered.

C. Model Establishment

For each dataset presented, the sampling frequency and SW size are 100 Hz and 100, respectively; therefore, the length of each data instance is 1s. We use 256 as the batch size and 100 as the number of epochs. The number of hidden units, learning rate, and RNN cell type are selected to be values with a minimum mean error for validation data. Hyperparameter tuning is conducted over the number of hidden units $l \in \{64, 128, 256\}$, learning rates $\in \{0.0001, 0.0005, 0.001\}$, and RNN types $\in$ {RNN, LSTM, GRU}. For both RAE and CRAE, the number of hidden units, learning rate, and RNN cell type are 256, 0.001, and long short-term memory (LSTM) cell, respectively. The major hyperparameters for each model are summarized in Table I.

The rectified linear unit is used as an activation function between hidden layers, and no activation is used for the final layer because this might limit the range of reconstructed values, which is not desirable for detecting anomalies with extreme values. All experiments were run in Python with the CUDA version 10.1 and PyTorch 1.7.1.

TABLE I
MAJOR HYPERPARAMETERS FOR THE BEST MODELS
OF EACH TYPE

Hyperparameters	**RAE**	**CRAE**
Sliding window size	100	
Batch size	64	
Epochs	100	
Hidden units	256	256
Learning rate	0.001	0.001
RNN cell type	LSTM	LSTM

TABLE Ⅱ
AUROC COMPARISON OF RAE AND CRAE

Abnormal Scenario	**RAE**	**CRAE**
Obstacles on a Straight Track	0.8753	0.9014
Increased Hoist TQ without an FOUP	0.4947	0.9083
Straight Driving + Curved State	0.4397	1.0000
Curved Driving + Straight State	0.5369	0.9983
Slide Operation + Straight State	0.4956	0.9953

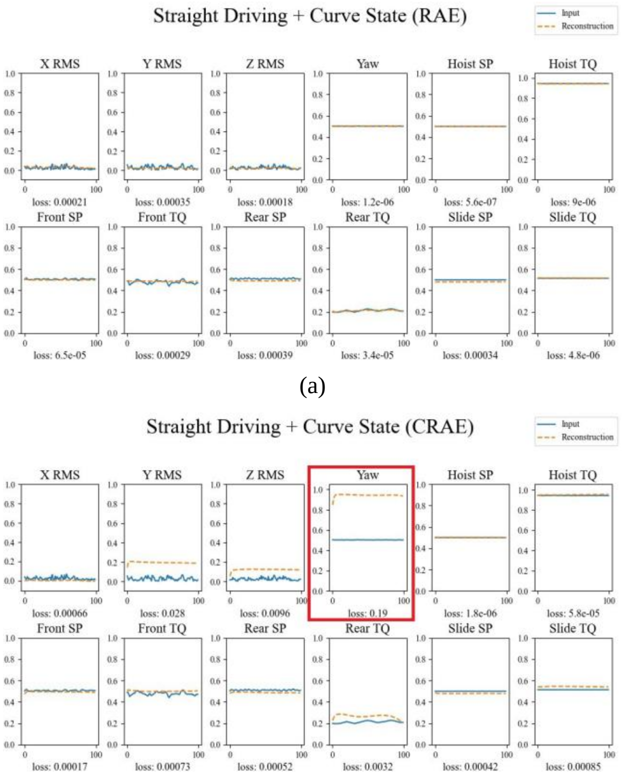
TABLE Ⅲ
AVERAGE LOSS
(ABNORMAL DATA ARE DENOTED AS *)

Abnormal Scenario	**RAE**	**CRAE**
Normal Straight Driving	0.00039	0.00043
Obstacles on a Straight Track*	0.00116	0.00128
Straight Driving + Curved State*	0.00037	0.02068
Normal Curved Driving	0.00183	0.00146
Curved Driving + Straight State*	0.00131	0.00339
Normal Hoist Operation	0.00035	0.00041
Increased Hoist TQ without FOUP*	0.00036	0.00125
Normal Slide Operation	0.00152	0.00158
Slide Operation + Straight State*	0.00152	0.01430

D. Performance Analysis

As the sensitivity control level may vary between FABs, the area under the receiver operating curve (AUROC) value, which is independent of a particular threshold, was used as a performance metric. The AUROC values of the RAE and CRAE are summarized in Table II. The AUROC value for each scenario was obtained relative to the relevant normal state. The average losses of each abnormal scenario and its relevant normal state are also presented in Table III for reference.

In all scenarios of anomaly detection, the CRAE model showed higher AUROC values than the RAE model (Table II). The AUROC difference was smaller in the Obstacles on a Straight Track scenario because the Yaw value of the sensor data can reflect the context of straight-track driving in the RAE model; however, the CRAE model showed a higher AUROC value by directly determining the context.



(a)

(b)

Figure 7. Visualization of input and reconstruction for **Straight Driving + Curved State** scenario by (a) the RAE model; and (b) the CRAE model.

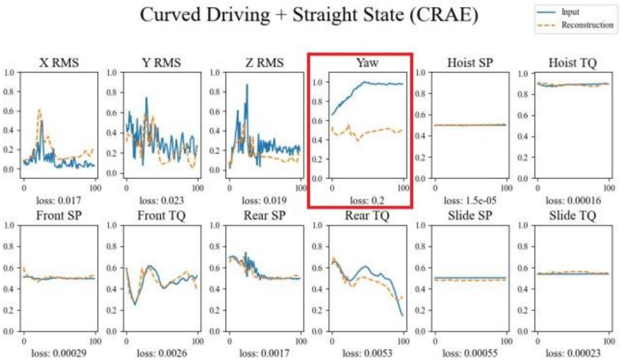


Figure 4. Visualization of input and reconstruction for **Curved Driving + Straight State** scenario by the CRAE model.

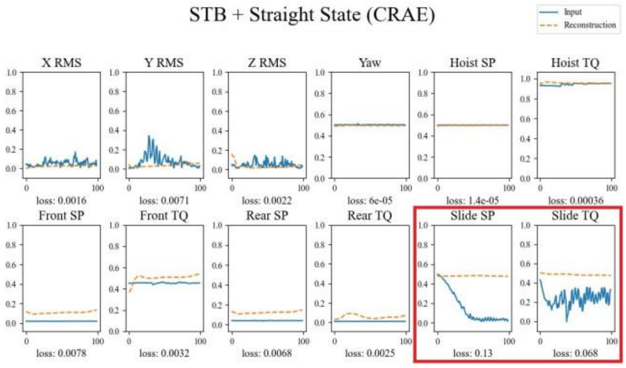


Figure 9. Visualization of input and reconstruction for **Slide Operation + Straight State** scenario by the CRAE model.

Moreover, the RAE model did not detect any state anomalies, because it did not consider any information about the states, whereas the CRAE model detected all state anomalies with very high AUROC values. To investigate the difference in detail, Fig. 7 illustrates an input sample (blue line) of the Straight Driving + Curved State data and the reconstructions (orange line) generated by the RAE and CRAE models. As can be seen, the RAE model did not detect this scenario as abnormal and reconstructed the data close to the input as normal straight-track driving data. In contrast, the CRAE model reconstructed the input data to match the assigned curved-track driving state, which resulted in a high reconstruction loss. In particular, the CRAE model reconstructed the Yaw value to be close to 1, which represented the state of curved-track driving. This shows that the CRAE model properly considered state information in learning the latent representation of the input data. Similarly, the CRAE model reconstructed the Curved Driving + Straight State data as straight-track driving by reconstructing Yaw to be close to 0.5 (static state) (Fig. 8). In the Slide Loading + Straight State scenario, the speed and torque of the slide motor were reconstructed as the static state by the CRAE model, which corresponds to the state of straight-track driving (Fig. 9).

E. Comparison with Conventional Methods

To further investigate the effectiveness of the CRAE structure in considering the additional contextual information, we compared the performance of our CRAE model with those of conventional anomaly detection methods: principal component analysis (PCA), k-nearest neighbors (KNN), feature bagging (FB), isolation forest (IF), and a one-class support vector machine (OC-SVM). As these conventional methods are designed to fit one-dimensional inputs, the aggregated observation $\mathrm{X_T} \in \mathbb{R}^{n \times T}$ is flattened into a one-dimensional vector as follows:

$$\mathrm{X} = [\mathrm{x}_1^1, \dots, x_T^1, x_1^2, \dots, x_T^2, \dots, x_1^n, \dots, x_T^n]^T \in \mathbb{R}^{nT \times 1} \quad (10)$$

We use the concatenated input of this flattened sensor vector and the state vector, $[\mathrm{X}\,;\mathrm{C}] \in \mathbb{R}^{(nT+k)\times 1}$ as the input of each method. The AUROCs are compared in Table IV. The CRAE model outperformed the other conventional methods. Therefore, the CRAE structure effectively considers additional contextual information rather than simply processing a combination of the flattened sensor vector and a state vector.

V. Conclusion

In modern FABs, semiconductor wafers are transported automatically on OHT systems. The failure of an OHT system interferes with the entire production line, and thus such systems must be continuously monitored so that anomalies can be rapidly detected and rectified. An OHT system also has various operating states, on which its behavior depends; thus, an OHT system's operating state should be considered in the process of system anomaly detection. Hence, we developed a CRAE model by incorporating a conditional structure into an RAE. To validate the performance of the CRAE model, we collected data for various abnormal scenarios in an actual OHT system test-bed. The CRAE model processed sensor data for these scenarios with consideration of the system state, and thus detected all state-dependent anomalies.

Future studies could employ various factors other than the states used in this study as the conditional input. For example, the tracks in FABs can be further subdivided according to their length, speed limits, and curvature. In addition, the identification number of OHT vehicles could be applied as a conditional input to address the individual characteristics of each vehicle. Furthermore, the CRAE model could be applied to other systems with multiple operational states, such as automated guided vehicle systems or stocker systems.

Acknowledgment

The authors thank the researchers and engineers at DAIM Research Corp. for their valuable comments and opinions on actual fab environments.

TABLE IV
Average Loss (Abnormal Data Are Denoted As *)

Abnormal Scenario	PCA	KNN	FB	IF	HBOS	OC-SVM	CRAE
Obstacles on a Straight Track	0.6961	0.6202	0.7936	0.7876	0.8048	0.6546	**0.9014**
Increased Hoist TQ without an FOUP	0.3789	0.8651	0.4598	0.394	0.4227	0.2415	**0.9083**
Straight Driving + Curved State	0.4550	0.7713	0.9515	0.4483	0.4016	0.6546	**1.0000**
Curved Driving + Straight State	0.4802	0.6620	0.5608	0.5421	0.4819	0.5144	**0.9983**
Slide Operation + Straight State	0.7760	0.7419	0.3727	0.6243	0.5351	0.3850	**0.9953**

References

[1] A. Zhakov, H. Zhu, A. Siegel, S. Rank, T. Schmidt, L. Fienhold, and S. Hummel, "Application of ann for fault detection in overhead transport systems for semi- conductor fab," IEEE Transactions on Semiconductor Manufacturing, vol. 33, no. 3, pp. 337–345, 2020.

[2] J. Park, J. Myung, M. Jo, S. Baek, and Y. Jang, "Autoencoder-based detection of anomalies in multiple time-series sensor signals for automated material han- dling systems of a semiconductor fabrication facility," submitted for publication.

[3] C. C. Aggarwal, "Outlier analysis," in Data mining. Springer, 2015, pp. 237–263.

[4] G. Jiang, P. Xie, H. He, and J. Yan, "Wind turbine fault detection using a denoising autoencoder with temporal information," IEEE/Asme transactions on mechatronics, vol. 23, no. 1, pp. 89–100, 2017.

[5] Y. Cui, P. Bangalore, and L. B. Tjernberg, "An anomaly detection approach based on machine learning and scada data for condition monitoring of wind turbines," in 2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS). IEEE, 2018, pp. 1–6.

[6] T. Junbo, L. Weining, A. Juneng, and W. Xueqian, "Fault diagnosis method study in roller bearing based on wavelet transform and

stacked auto-encoder," in The 27th Chinese Control and Decision Conference (2015 CCDC). IEEE, 2015, pp. 4608–4613.

[7] H. Liu, J. Zhou, Y. Zheng, W. Jiang, and Y. Zhang, "Fault diagnosis of rolling bearings with recurrent neural network-based autoencoders," ISA transactions, vol. 77, pp. 167–178, 2018.

[8] Z. Chen and Z. Li, "Fault diagnosis method of rotating machinery based on stacked denoising autoencoder," Journal of Intelligent & Fuzzy Systems, vol. 34, no. 6, pp. 3443–3449, 2018.

[9] Y. Huang, C.-H. Chen, and C.-J. Huang, "Motor fault de- tection and feature extraction using rnn-based variational autoencoder," IEEE Access, vol. 7, pp. 139 086–139 096, 2019.

[10] X. Wu, Y. Zhang, C. Cheng, and Z. Peng, "A hybrid classification autoencoder for semi-supervised fault di- agnosis in rotating machinery," Mechanical Systems and Signal Processing, vol. 149, p. 107327, 2021.

[11] D. Ramotsoela, A. Abu-Mahfouz, and G. Hancke, "A survey of anomaly detection in industrial wireless sensor networks with critical water system infrastructure as a case study," Sensors, vol. 18, no. 8, p. 2491, 2018.

[12] D. J. Atha and M. R. Jahanshahi, "Evaluation of deep learning approaches based on convolutional neural net- works for corrosion detection," Structural Health Moni- toring, vol. 17, no. 5, pp. 1110–1128, 2018.

[13] T.-Y. Kim and S.-B. Cho, "Web traffic anomaly detec- tion using c-lstm neural networks," Expert Systems with Applications, vol. 106, pp. 66–76, 2018.

[14] T. Chen, X. Liu, B. Xia, W. Wang, and Y. Lai, "Un- supervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder," IEEE Access, vol. 8, pp. 47 072–47 081, 2020.

[15] M. Mirza and S. Osindero, "Conditional generative ad- versarial nets," arXiv preprint arXiv:1411.1784, 2014.

[16] O. M. Ezeme, Q. H. Mahmoud, and A. Azim, "Design and development of ad-cgan: Conditional generative ad- versarial networks for anomaly detection," IEEE Access, vol. 8, pp. 177 667–177 681, 2020.

[17] R. Ahsan, W. Shi, X. Ma, and W. Lee Croft, "A comparative analysis of cgan-based oversampling for anomaly detection," IET Cyber-Physical Systems: Theory & Applications, 2021.

[18] R. Gayathri, A. Sajjanhar, Y. Xiang, and X. Ma, "Multi- class classification based anomaly detection of insider activities," arXiv preprint arXiv:2102.07277, 2021.

[19] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative mod- els," Advances in neural information processing systems, vol. 28, pp. 3483–3491, 2015.

[20] A. A. Pol, V. Berger, C. Germain, G. Cerminara, and M. Pierini, "Anomaly detection with conditional variational autoencoders," in 2019 18th IEEE interna- tional conference on machine learning and applications (ICMLA). IEEE, 2019, pp. 1651–1657.

[21] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional variational autoencoder for predic- tion and feature recovery applied to intrusion detection in iot," Sensors, vol. 17, no. 9, p. 1967, 2017.

[22] L. Yang and Z. Zhang, "A conditional convolutional autoencoder-based method for monitoring wind turbine blade breakages," IEEE Transactions on Industrial Infor- matics, vol. 17, no. 9, pp. 6390–6398, 2020.