An Encoder-Decoder Based Approach for Anomaly Detection with Application in Additive Manufacturing

Baihong Jin[†1] Yingshui Tan[†1,2] Alexander Nettekoven[*3] Yuxin Chen[*4] Ufuk Topcu[3]
Yisong Yue[4] Alberto Sangiovanni-Vincentelli[1]
[1]Department of EECS, University of California, Berkeley, USA
[2]RWTH Aachen University, Germany
[3]University of Texas at Austin, USA
[4]California Institute of Technology, USA

Abstract—We present a novel unsupervised deep learning approach that utilizes the encoder-decoder architecture for detecting anomalies in sequential sensor data collected during industrial manufacturing. Our approach is designed not only to detect whether there exists an anomaly at a given time step, but also to predict what will happen next in the (sequential) process. We demonstrate our approach on a dataset collected from a real-world Additive Manufacturing (AM) testbed. The dataset contains infrared (IR) images collected under both normal conditions and synthetic anomalies. We show that the encoder-decoder model is able to identify the injected anomalies in a modern AM manufacturing process in an unsupervised fashion. In addition, it also gives hints about the temperature non-uniformity of the testbed during manufacturing, which is what we are not aware of before doing the experiment.

Index Terms—additive manufacturing, anomaly detection, fault detection

I. INTRODUCTION

Anomaly detection is an important technique that serves as the basis of applications across a diverse variety of domains, such as fault detection, intrusion and fraud detection [18], and process control. The goal of anomaly detection is to identify patterns in data that do not conform to a well-defined notion of normal behavior [2]. Early detection of anomalies and faults allows us planning preventive maintenance for model manufacturing, and thus it is crucial for process control.

The availability of massive amount of data due to the introduction of pervasive sensing techniques has brought plenty of opportunities for data-driven anomaly detection applications; however, an unresolved challenge is how to make use of these data for anomaly detection, especially when there is no label

information that can be used to differentiate between normal and anomalous working conditions.

A. Learning-Based Anomaly Detection

Depending on the availability of labeled anomalous data, learning-based anomaly detection approaches can generally be categorized into *supervised* and *unsupervised* methods. Supervised methods utilize label information for both normal and anomalous data to train classification models. The trained classification models from supervised learning can not only tell the existence of faults but also indicate the likelihood of an input belonging to a particular type of fault.

A review of the literature reveals that data-driven approaches relying on supervised learning have demonstrated promising results in various applications, e.g. Fault Detection and Diagnosis (FDD) in air conditioning systems [9], [13], [14].

To train a well-performing model using supervised learning, a good amount of labeled data from both normal and anomalous conditions are needed, which is not always easy to obtain in practice.

In addition, supervised models typically lack the ability to identify an unseen example that does not belong to any of the classes that appear in the training set. In the context of anomaly detection, models trained with supervised learning are likely to give incorrect predictions on out-of-distribution data instances. This is a limitation of supervised methods because it is almost impossible to obtain every possible type of anomaly that could happen on a system. To address this problem, Jin et al. recently proposed a FDD method that uses Monte-Carlo dropout [9] to estimate the prediction uncertainty of deep neural networks. The method was applied to the identification of incipient faults that are not represented in the training data that only consists of labeled data of normal and severe faults.

In scenarios where labeled *anomalous* data are scarce or unavailable, unsupervised and semi-supervised anomaly detection approaches are usually applied, because only normal data are required to train a detection model. The two approaches differ in their assumptions about the labels of training data. In

semi-supervised learning, it is assumed that the training set is comprised of only data instances from the normal class[1], while in an unsupervised setting, it is often implicitly assumed that few anomalous instances can exist in the training data [2]. We note that the approach we introduce in this paper can apply to both settings. We choose to use the term "unsupervised learning" throughout this paper to refer to both situations where normal data account for the majority or the entirety of the training data. Although unsupervised approaches usually lack the discriminative ability to assign labels to anomalous data, it is still considered an appealing complement to supervised approaches in many real-world applications.

Recently, neural network approaches, especially deep neural networks, have attracted much attention from the machine learning community, due to their ability to process natural data in their raw form and learn internal representations that can be used for detecting or classifying patterns [11]. Yet, as the authors of the recent review paper [11] also pointed out, supervised learning accounted for the majority of the recent success of deep learning, while unsupervised learning is expected to be far more important in the longer term. This paper aims at taking advantage of the recent development of deep learning and provide a methodology for developing unsupervised anomaly detection algorithms for handling sequential sensing data in industrial applications.

B. Our Contributions

In particular, we investigate the applicability of an encoder-decoder approach on sequential image sensing data collected in a real industrial setting. The contributions of this paper are two-fold:

- We propose using an encoder-decoder architecture for detecting anomalies in sequential image sensing data collected from AM process. The learning process is unsupervised, meaning that no anomalous data are needed *a priori* to train the detection model.
- We design a Convolutional Neural Network (CNN)-based encoder-decoder network to monitor the manufacturing process of the Laser Additive Manufacturing Pilot System (LAMPS) testbed, a platform that uses Selective Laser Sintering (SLS) technology for AM. In our experiment, the network can not only detect the artificially injected laser anomalies with high accuracy, but also can indicate regions of the manufacturing testbed where the temperature is higher than usual. Our results demonstrate the effectiveness of the proposed algorithm in detecting anomalous phenomena.

C. Paper Organization

The remainder of this paper is organized as follows. In Sec. II, we will give the background about LAMPS, the encoder-decoder architecture and deep-learning-based anomaly detection approaches. We will define the anomaly

---

[1]Note that semi-supervised anomaly detection differs from the traditional notion of "semi-supervised learning" in machine learning, where both label and unlabeled data are used simultaneously for training.

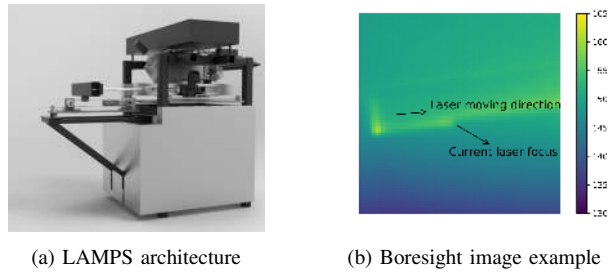(a) LAMPS architecture (b) Boresight image example

Fig. 1: (a) LAMPS testbed [17]. (b) An example image captured by the boresighted temperature sensor.

detection problem for sequential data in Sec. III. We will describe our anomaly detection methodology for sequential image data in Sec IV. In Sec. V, we will describe in details our anomaly detection algorithm when applied to a real-world AM dataset with injected faults. Experimental results will be demonstrated and evaluated in Sec. VI. We will discuss future work and conclude the paper in Sec. VII.

II. BACKGROUND

A. Laser Additive Manufacturing Pilot System (LAMPS)

AM technologies have transformed the manufacturing landscape. [6] In contrast to traditional manufacturing technologies, AM technology is capable of printing 3D parts with highly complex geometries in a single process step. Due to its versatility, AM technology are used in a wide variety of applications such as medical devices and aircraft manufacturing [1]. One of the prominent AM technologies is Selective Laser Sintering (SLS) that uses a laser to form solid parts out of powdered material. Building parts with consistent high-quality is a key challenge for the SLS process today [6]. Therefore, having an algorithm that can monitor the SLS printing process and can indicate potential anomalies will significantly improve SLS process control. This, in turn, will lead to improved part quality and ensure repeatability.

We now briefly introduce the SLS printing process and the testbed we used for data collection and testing purposes. SLS utilizes a laser to fuse powder geometries layer-by-layer and hereby generates a solid 3D structure. At the beginning of each layer, a roller spreads a new powder layer across the powder bed. Once the powder has been spread, the laser melts the cross-section of the desired part according to the digital 3D model. After the laser has finished scanning for the current layer, a new powder layer is spread and the scanning process is repeated. Over time the melted powder locations on each layer will cool down and will solidify to one.

LAMPS is a SLS testbed that was designed and built for process control research. LAMPS is capable of building 3D parts out of high-performance plastics (melting temperatures as high as 350 °C) and is equipped with a variety of sensors, such as IR and visual cameras, that provide *in-situ* measurement access. Fig. 1a shows the general architecture of the LAMPS testbed.

In the context of this paper, we focus on the high-speed mid-wave infrared (IR) camera which is bore-sighted with the laser optics to record the laser focus and its immediate surrounding. The camera has a resolution of 64×64 pixels and has a maximum recording frame rate of 2.24 kHz. Fig. 1b shows an exemplary IR image of the bore-sighted camera. The recorded IR information is translated to gray-scale (single-channel) temperature images, where the intensity value of each pixel represents the measured temperature value at that pixel.

B. Encoder-decoder architecture

The encoder-decoder architecture has proven to be a useful approach for learning (deep) representations, and is widely used in various application domains of deep learning, including machine translation [3], and image denoising [23]. An encoder-decoder model generally consists of three parts: the encoder, the latent space representation, and the decoder. The purpose of the encoder network Enc is to transform the input data into a latent space representation z that is often a vector; the decoder network Dec then produces the output by decoding z. During training, the encoder and the decoder are trained together to minimize the empirical risk.

Proper design of the latent space representation z is crucial to the successful application of the encoder-decoder approach. Let us take the basic autoencoder model as an example. An autoencoder is a neural network model that is trained to reconstruct its input. In other words, an autoencoder is trained to learn an identity function for the data distribution. By constraining z to be a low dimensional vector, the training process encourages the model to learn the most useful information for reconstructing the input.

C. Unsupervised anomaly detection with deep learning

Supervised deep learning has been extensively studied in various applications domains.

In fault/anomaly detection tasks, we often do not have access to the entire spectrum of off-nominal data, as well as the labels that come along with it. As a result, unsupervised approaches that do not require labeled anomaly data are more suitable in such scenarios.

In this paper, we aim to explore unsupervised anomaly detection using a deep learning approach. Specifically, we will adopt the encoder-decoder scheme described earlier in Sec. II-C.

The general idea behind unsupervised anomaly detection approaches is to find an approximate model that can capture the normal behavior of complex systems. The approximate model can then be used to flag anomalies if the deviation of the predicted behaviors of the trained model from the actual observation exceeds some certain threshold. Examples that share this general idea include One-class Support Vector Machine (OC-SVM) [5], [8], [20], Principal Component Analysis (PCA) [12] and autoencoders [19].

The encoder-decoder schemes for anomaly detection that appeared in literature in general fall into three categories, which differ in their prediction outputs: 1) autoencoder models [15], 2) prediction models, and 3) composite models [22] that performs both reconstruction and regression. We denote our observation at time instant τ by S_τ. The observations we observe in time then forms as a sequence $\{S_\tau\}$. Let g be a function that maps an input sequence of length p to an output sequence of length q. These encoder-decoder schemes are therefore summarized below:

$$
(S_{\tau_{-p+1}}, \ldots, S_{\tau_0})
$$
$$
\xrightarrow{g}
\begin{cases}
(S_{\tau_{-p+1}}, \ldots, S_{\tau_0}), & \text{reconstruction model,} \\
(S_{\tau_1}, \ldots, S_{\tau_q}), & \text{regression model,} \\
(S_{\tau_{-p+1}}, \ldots, S_{\tau_q}), & \text{composite model.}
\end{cases}
\tag{1}
$$

As previously described, reconstruction models (a.k.a. autoencoders) aim to find a compact representation for input data distribution. Depending on the format of the input data, different neural network architectures or their combinations are used to design encoders and decoders. Autoencoders are first trained on data that are normal or almost fault-free. The reconstruction errors given by autoencoder models are often used as anomaly scores to indicate potential anomalies. This approach is seen in previous literature for anomaly detection in multivariate timeseries [15].

Similarly, we can also use the encoder-decoder architecture for prediction tasks. In the case of time series data, a neural network prediction model can be trained to predict the future from past observations. Taking the past p observations as input $(S_{\tau_{-p+1}}, \ldots, S_{\tau_0})$, the model is trained to predict the next q observations $(S_{\tau_1}, \ldots, S_{\tau_q})$. During training, the encoder will look for information needed for the decoder to predict the future, and encode the information as latent space representations. In this case, the prediction errors are used to indicate potential anomalies.

The authors of [22] argue that a composite model, by performing the reconstruction and the regression tasks simultaneously, can overcome the drawbacks of each one when performed alone, and thus achieving better performance at learning useful representations in the data. Previous literature reports on schemes for detecting anomalies in videos [16] and multivariate time series. In our case study to be later discussed, we designed our encoder-decoder model as a composite model to leverage the advantages of both reconstruction and regression models.

III. THE ANOMALY DETECTION PROBLEM

Assume that we are given a series of observation data, $S_{\tau_0}, S_{\tau_1}, \ldots, S_{\tau_i}, \ldots$, where each $S_{\tau_i} \in \mathcal{S}$ ($\mathcal{S}$ being the input domain) denotes the representation of the ith data point in the sequential data. In the anomaly detection setting, we assume that all data points from the training set are in the normal state.

Let $\mathcal{F}$ be a model class, where each $f \in \mathcal{F} : \mathcal{S} \to \mathbb{R}_{\geq 0}$ denotes a score/fitness function that characterizes how close a data point is to an abnormal state, i.e., larger f implies higher chance of a data point being abnormal.

For a given threshold value $\epsilon > 0$, we define the detection precision of f as

$$\text{prec}(f, \epsilon) = \mathbb{E}_S[\mathbb{1}\{S \text{ is abnormal}\} \mid f(S) > \epsilon]$$

where the expectation is taken over the distribution of the test data, and the recall of f as

$$\text{recall}(f, \epsilon) = \mathbb{E}_S[\mathbb{1}\{f(S) > \epsilon\} \mid S \text{ is abnormal}]$$

Our goal is to learn a score function $f_{\text{score}} \in \mathcal{F}$ and a corresponding threshold ϵ, such that $(f_{\text{score}}, \epsilon)$ achieves the best detection accuracy and recall of anomalies on the (unseen) test data.

IV. METHODOLOGY

We utilize the encoder-decoder architecture described in Sec. II-C to design a neural network that can be used to detect possible anomalies in AM process. Since we are dealing with image data in LAMPS application, we choose to use CNNs [10] as the main building blocks for our encoder-decoder model. Our approach uses the composite prediction model described earlier in Sec. II-C – the designed model will not only attempt to reproduce the input but also predict what will happen next.

A. CNN-based encoder-decoder model

In our unsupervised learning setting, we only have access to data points collected under normal condition. The learning goal is to use a neural network to model the normal behaviors of the system under study. Outliers to the learned distribution will be identified as potential anomalies.

Let us suppose that each observation S_τ in the sequential data is a single-channel 2D image of dimension $m \times n$, i.e. $S_\tau \in \mathbb{R}^{m \times n}$. To capture the temporal correlations among the observations, a sliding window approach can be used to divide the original image sequence into *snippets*, where each snippet $\mathbf{Z}_k \in \mathbb{R}^{m \times n \times (p+q)}$ comprise of $p + q$ consecutive frames, and k is the index of the snippet.

When a regression or composite prediction scheme is used to train an encoder-decoder model, the frames in a snippet constitute the input and the output. For training a regression model, the first p frames in a snippet $\mathbf{X}_k \in \mathbb{R}^{m \times n \times p}$ constitute the model input, and the rest q frames are the output to be predicted. In the case of a composite model, the input is still the $\mathbf{X}_k$, and the output is the entire $p + q$ frames. If we view the frames in a snippet as channels in an image, the learning problem can be cast as an *image-to-image* translation task. To be more specific, we will train the encoder-decoder network $\mathcal{M}$ to learn a mapping $g : \mathbb{R}^{m \times n \times p} \to \mathbb{R}^{m \times n \times (p+q)}$ that transforms a p-channel image input $\mathbf{X}_k$ to an output $\hat{\mathbf{Z}}_k$ with $(p+q)$ channels. The prediction output $\hat{\mathbf{Z}}_k$ can be seen as the combination two parts, $\hat{\mathbf{X}}_k$ and $\hat{\mathbf{Y}}_k$. $\hat{\mathbf{X}}_k$ is the reconstruction of the p input frames, and $\hat{\mathbf{Y}}_k$ is a prediction of the q frames following the input frames.

When training the encoder-decoder model, we aim to minimize the errors on both the reconstruction part and the regression part. Since the model input and output are both

images, the following pixel-wise Mean Square Error (MSE) can be used as the error metric on frame S_τ.

$$\ell_{mse}(S_\tau, \hat{S}_\tau) = \|S_\tau - \hat{S}_\tau\|_F, \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix.

Let us suppose the frames in snippet k are taken at time instants $\tau_k^0, \tau_k^1, \ldots, \tau_k^{p+q-1}$. By choosing (2) as the error metric, we can define the reconstruction error e_k^{rec} and regression error e_k^{reg} on snippet k as follows

$$e_k^{\text{rec}} \doteq \sum_{0 \le i < p} \ell_{mse}(S_{\tau_k^i}, \hat{S}_{\tau_k^i}), \tag{3}$$

$$e_k^{\text{reg}} \doteq \sum_{0 \le i < p+q} \ell_{mse}(S_{\tau_k^i}, \hat{S}_{\tau_k^i}). \tag{4}$$

The loss function L to minimize during model training can then be defined as as the weighted sum of reconstruction error e_k^{rec} and regression error e_k^{reg} on all training samples $k \in \mathcal{K}$.

$$L = \sum_k e_k^{\text{rec}} + \lambda e_k^{\text{reg}} \tag{5}$$

where λ is a weighting factor that adjusts the relative importance between the reconstruction error and the regression error.

B. Using the trained model for anomaly detection

Assuming the trained encoder-decoder model has learned a good representation of the normal behavior of the system, the differences between the predicted images and their corresponding ground truth can be used to indicate possible anomalies. By comparing the images, we are essentially getting a large number of pixel-wise errors, and thus a method is needed to process this information in order to detect and locate the anomalies.

One simple idea is to use the original loss function (5) that we used for training the network. These loss values can be derived directly from the prediction results, and can be used as good indicators for evaluating the network's prediction quality; however, this approach also suffers from two drawbacks. First, if the anomaly is only localized to a small area, it is likely that the prediction errors are only significant in a small part of the image. When we calculate the pixel-wise MSE over the entire image, useful indications of anomalies may be buried in noise and averaged out. In addition, even if a significant loss is observed on an image, this approach only indicates a potential anomaly on the image level, but it does not give further hint about the occurrence of this anomaly. It is unknown whether the anomaly is local to only a small area or affects the entire image.

To address the above mentioned challenge when using for evaluating image-to-image differences, we propose a "spatial scoping" approach: we aim to find a $a \times b$ window from the $m \times n$ error matrix $E_\tau = S_\tau - \hat{S}_\tau$ that has the maximum Frobenius norm. The new error metric ℓ_{ss} is thus defined as:

$$\ell_{ss}(S_\tau, \hat{S}_\tau) = \max_{\substack{0 \le i \le m-a \\ 0 \le j \le m-b}} \|E_\tau^{i:i+a, j:j+b}\|_F, \tag{6}$$
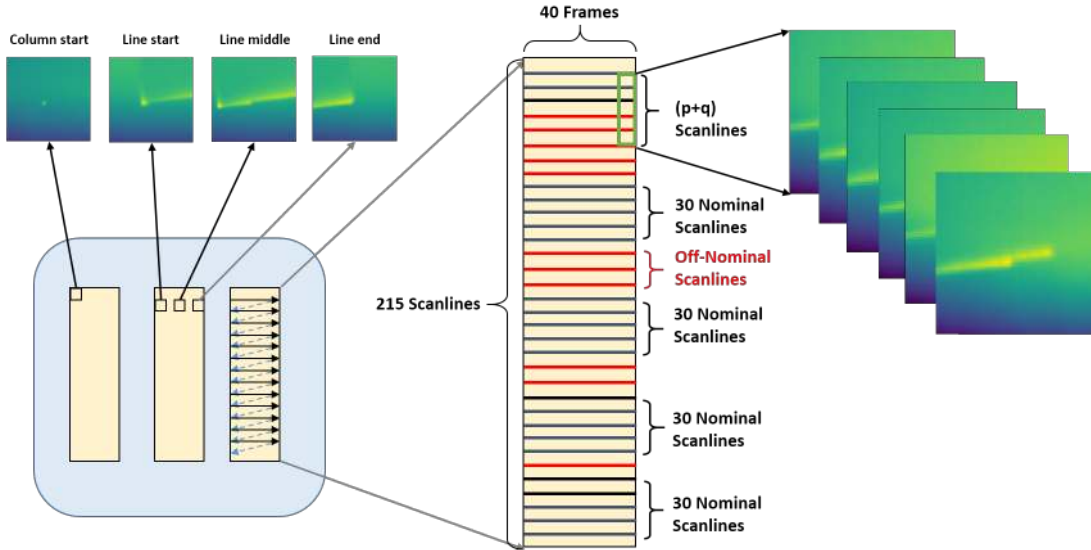
Fig. 2: An top-down illustration of the benchmark dataset. (Left) On the top we show examples of boresight images when the laser head it at column start, line start, line middle and line end, respectively; on the bottom we illustrate the trace of scanning laser; (Middle) the nominal and off-nominal scan lines pattern; (Rigth) Examples of preprocessed data (boresight images)

where the superscript in $E_\tau^{i:i+a,j:j+b}$ indicates the position of the window in the original error matrix E_τ.

Next we define two anomaly scores, the *reconstruction anomaly score* $f_{\text{score}}^{\text{rec}}(\tau)$ and the *regression anomaly score* $f_{\text{score}}^{\text{reg}}(\tau)$, as the metrics for evaluating the "degree of anomaly" of an observation S_τ. Note that there is more than one snippet that encompasses S_τ because we used a sliding window approach to generate the snippets. To get a single anomaly score taking into account the prediction errors from all relevant snippets, we define the anomaly score as the average prediction errors from all these snippets.

Suppose K_τ^{rec} is the index set of snippets whose reconstruction window covers S_τ, and K_τ^{rec} is the index set of snippets whose reconstruction or regression window covers S_τ. The reconstruction and regression anomaly scores on S_τ can be defined by as follows

$$f_{\text{score}}^{\text{rec}}(\tau) = \frac{1}{|K_\tau^{\text{rec}}|} \sum_{k \in K_\tau^{\text{rec}}} e_k^{\text{rec}}, \quad \text{reconstruction} \tag{7a}$$

$$f_{\text{score}}^{\text{reg}}(\tau) = \frac{1}{|K_\tau^{\text{reg}}|} \sum_{k \in K_\tau^{\text{reg}}} e_k^{\text{reg}}, \quad \text{regression} \tag{7b}$$

where $|K_\tau^{\text{rec}}|$ and $|K_\tau^{\text{reg}}|$ are the cardinalities of sets K_τ^{rec} and K_τ^{reg} respectively. A notable difference between K_τ^{rec} and K_τ^{reg} is their sizes. In the reconstruction case, all snippets whose regression windows (of length p) cover S_τ are included in K_τ^{rec}. As a result, $|K_\tau^{\text{rec}}| = p$, except at the start or end of sequence $\{S_\tau\}$ because at the boundaries there are be fewer snippets covering an observation. In the regression case, as long as an anomaly is seen in either the reconstruction window (of length p) or the regression window (of length q), the anomaly would (probably) be caught in the regression error.

Therefore, $|K_\tau^{\text{reg}}| = p+q$ except at the start or end of sequence $\{S_\tau\}$.

The anomaly scores introduced above can be used to evaluate how likely an observation S_τ will correspond to an anomalous state of the system under study. Later in Sec. V and Sec. VI, we will present a case study on LAMPS to illustrate our proposed approach.

V. ALGORITHMIC DETAILS FOR LAMPS

A. Benchmark dataset with synthetic faults

Fig. 2 shows the laser trajectory in LAMPS machine. It is clear that the laser follows a periodical motion pattern in our experiment. The laser firstly moves rightward till the right boundary of the column and then move leftward to the left boundary. During each period of motion, the laser power will move forward 1 unit in the line axis. There is no laser power in the leftward process which is depicted with dashed lines in Fig. 2. We therefore only took the rightward process into consideration in this experiment.

For testing our anomaly detection algorithm, we created an "off-nominal" build with the LAMPS machine. During this build, the laser power was altered at specific time instances from its nominal power. We will detail the layout of the build below.

The off-nominal build consisted of three columns being built over the course of 250 layers. Each column had the same off-nominal pattern applied in order to create a large dataset. For each layer, the laser scanned the rectangles (the horizontal section of the columns) with straight scan lines that were horizontally aligned. Off-nominal conditions were applied to every fourth layer by scanning specific scan lines with off-nominal laser power instead of nominal laser power. The off-nominal conditions were only applied every fourth layer to
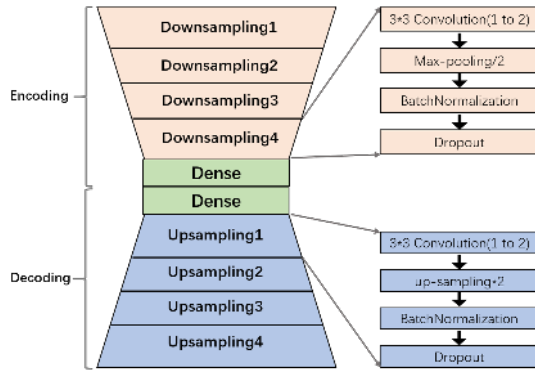
Fig. 3: Network Structure

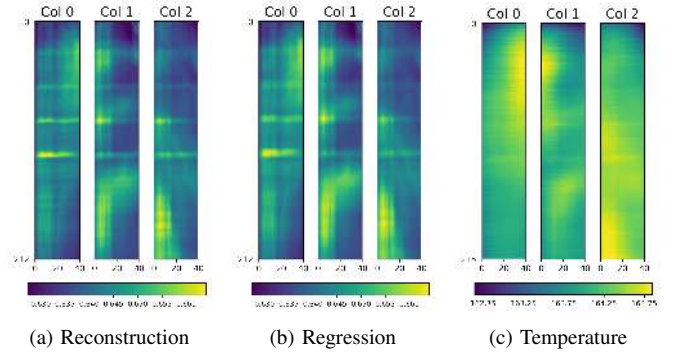(a) Reconstruction   (b) Regression   (c) Temperature

Fig. 4: Top-down views

TABLE I: Experiment results on different layers

Layers	Off-nominal laser power (% of max value)	Absolute power deviation (% of max value)	Precision	Recall
A1	58	13	0.93	0.95
A2	56	11	0.90	0.99
A3	54	9	0.88	0.87
A4	50	5	0.81	0.65
A5	48	3	0.75	0.61

ensure that there would be no temperature influences between off-nominal layers.

The horizontal cross-section of each column is of a rectangle shape. Fig. 2 illustrates the off-nominal scan line pattern for one of the three rectangles. Each rectangle consists of 215 horizontal scan lines and the bore-sight camera is able to approximately take 40 frames for each scan line. The off-nominal laser power magnitude stayed the same within every off-nominal layer, but was continuously changed throughout the build. For more comprehensive testing, the anomalies injected have different lasting areas, from 1 line to 4 lines.

B. CNN-based encoder-decoder network design

To create training data for the encoder-decoder model, we divide the dataset into snippets with each consisting $p + q$ frames. As illustrated in Fig. 2, each snippet is oriented in the vertical direction and spans $p + q$ scan lines. The frames within a scan line are ordered by the time sequence the images were taken. Let us denote by $S_{i,j}$ the jth frame taken in the ith scan line; here the indices for column number and layer number are omitted for brevity. The encoder-decoder model is trained to transform each input data point $\mathbf{X}_k = (S_{i,j}, S_{i-1,j}, \ldots, S_{i-p+1,j}) \in \mathbb{R}^{m \times n \times p}$ into a predicted output $\hat{\mathbf{Z}}_k = (\hat{S}_{i,j}, \hat{S}_{i-1,j}, \ldots, \hat{S}_{i-p+1,j}) \in \mathbb{R}^{m \times n \times (p+q)}$.

We choose a VGG-based [21] structure. The convolution kernel size is chosen to be 3×3 and the pooling kernel size is chosen to be 2×2 to build a deeper network instead of using a large kernel size. As shown in Fig. 3, in our network there are four stacked down-sampling layer groups and four stacked up-sampling layer groups to sample the data and reconstruct the data respectively. Each down-sampling group has one or two Convolution layers (depending on the network depth) and a "Maxpooling" layer and correspondingly each up-sampling layer has the same number of "Convolution" layers and a "Up-sampling" layer. Functionally, when the data is input into the neural network, each down-sampling layer group will down-sample the spatial dimensions (width, height) and double the depth of the data while each up-sampling layer will up-sample the spatial dimensions and halve the depth. Between the up-sampling groups and down-sampling groups, we set two fully connected layers, from which the latent

space representations can be extracted. In addition, in order to prevent the network from over-fitting, we add a "Dropout" layer to each group. To make training more efficient, we add a "BatchNormalization" [7] layer to each group between the "Maxpooling" layer and the "Dropout" layer.

VI. EXPERIMENTAL EVALUATION

A. Data preparation and preprocessing

We choose five nominal layers as our training set and five off-nominal layers with different size of anomalies as our test set; see Table I for details about the off-nominal layers.

In our data preprocessing step, the data was normalized by changing the range (the difference between max and min values) of the data to one. We down-sampled the original 64×64 image data to a resolution of 32×32, to reduce the complexity of our network. We chose $p = q = 3$ in our experiment for creating the snippets as described in Sec. V-B.

To improve the robustness of our model against small perturbation in input data, we augmented our dataset by adding a small Gaussian noise (with zero mean and a standard deviation of $0.01°$C) to the training data. In addition, we know from physics that the thermodynamics of the powder bed is largely governed by the gradient of the temperature distribution; therefore, we can generate additional synthetic data by adding a constant temperature bias b (in our experiment $b \in [-1.8°C, +1.8°C]$) to the original data. This helps regularize our network to better capture the underlying thermodynamics.

B. Network implementation and hyperparameter tuning

We used Keras [4] as the framework for implementing the encoder-decoder model. λ was set to 1 so that the reconstruction and regression errors were equally weighted. The model was trained for 500 epochs.
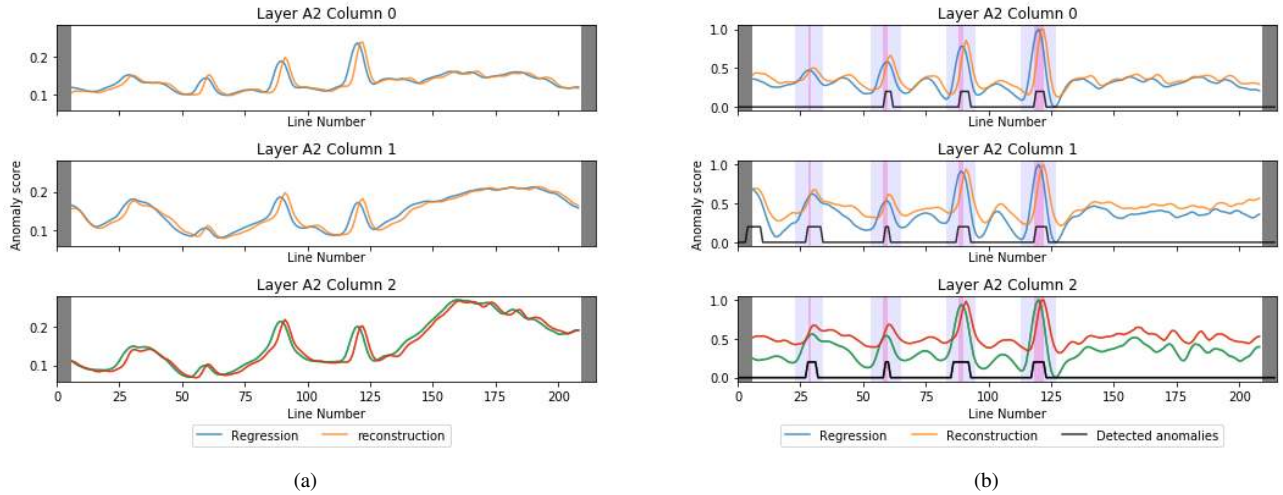
Fig. 5: (a) The line-wise reconstruction and regression anomaly scores, averaged on each scan line, and (b) the de-trended and normalized line-wise reconstruction and regression anomaly scores, as well as the detected anomalies. In (a), the raw MSE error metric (2) is used to calculate the errors and the anomaly scores; in (b) the spatial scoping error metric (6) is used. To give the readers a clearer understanding, we use dark pink shades in (b) to indicate the locations of injected anomalies. A lighter pink color is used to indicate lines adjacent to the injected anomalies if one (or more) window that is used for calculating the anomaly score at this line overlaps with the injected anomalies. In our setting, the affected region has a width of $p + q - 1 = 5$ on each side of an anomaly. In the plots, the first (and last) three lines are grayed out to ignore boundary effects.
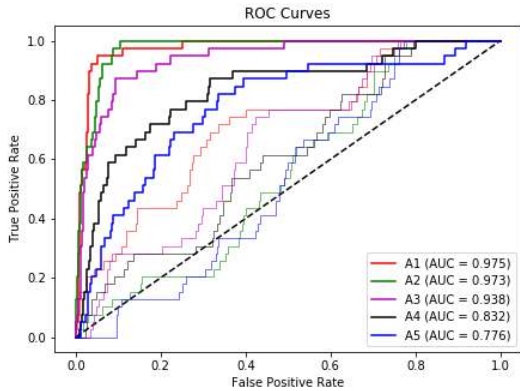

Fig. 6: The ROC curves of our learning-based model (darker colors) vs. the non-learning model (lighter colors) on the five test layers A1-A5 with injected anomalies.

In our unsupervised setting, we did not have labeled anomalous data for using the conventional cross-validation technique to tune the hyper-parameters as in supervised learning. To create a validation set that contains both normal and anomalous data points, we used a simple method to create some synthetic anomalies. 20% of the training data was first picked out as the validation data. We tuned the hyper-parameters (e.g., dropout rates and the number of convolution kernels) until the validation loss on normal data had converged.

C. Result analysis and evaluation

To assess the performance of our model, we tested our model using layers that have different degrees of anomaly. We plotted the distributions of the reconstruction anomaly score in Fig. 4a and the regression anomaly score in Fig. 4b from a top-down view, with the raw MSE (2) used as the error metric. It can be clearly seen that some of the regions with a high anomaly score are located along certain scan lines. To better visualize and quantify the variation of anomaly scores across different scan lines, we display in Fig. 5a the average anomaly score along each scan line. From the plot, we can see that scan lines with high anomaly scores appear either as "sharp peaks", or as "big bumps", which may indicate different causes of anomalous conditions, and should thereby be treated and analyzed separately.

Sharp peaks To isolate these sharp peaks in Fig. 5a, a detrend technique can be applied to filter out the slowly varying component; here we use a simple detrend technique by subtracting the signal using a window size of 20. In addition, we also apply the spatial scoping technique (6) as the error metric for calculating the anomaly score. The resulting signal is displayed in Fig. 5b, where we can see a clear correlation between the large peaks and the injected anomalies. A simple thresholding method is used to do anomaly detection. We test our model on the entire training sets and choose the maximum line-wise regression score as the threshold, so no false positives can be detected on all these normal layers. Most injected anomalies can now be correctly detected. We also observe that the first anomaly is difficult to detect. Due to the

fact that the deviated laser power merely lasts for 1 line, the temperature there has not yet been significantly changed since the energy accumulated is not sufficient to cause temperature variation in case of limited lasting time and these anomalies may sometimes be buried in the noise. Table I shows the performance of our detection model on the five off-nominal layers. The detection model performs better on layers with higher laser power deviation; the precision and recall rates both exceed 90% in layers A1 and A2. In layers A4 and A5, the precision and recall rates drop significantly, which is due to the reduced temperature disturbance due to smaller laser power deviation. It can also be seen in Fig. 6, the Area Under the Curve (AUC) rate exceeds 0.97 in layer A1 and layer A2 but drops to only 0.776 in layer A5.

For comparison, we also implemented a simple non-learning method that simply used raw temperature measurement for detecting anomalies. The anomaly score for each image S_τ is defined as its root-mean-square intensity (temperature) value over the image, i.e.

$$f_{\text{score}}^{\text{temp}}(\tau) = \|S_\tau\|_F. \tag{8}$$

As an example, we visualize the distribution of the anomaly scores on Layer A2 from the non-learning method in Fig. 4c. The ROC curves obtained from applying this non-learning method on these anomalous layers are displayed in Fig. 6. It can be seen that our encoder-decoder approach gives a much superior performance to the non-learning method.

Big bumps Having found the cause of sharp peaks, we would like to identify the cause of large bumps in the anomaly scores. By comparing the three top-down views in Fig. 4, we observe an obvious correlation among these large bump regions. We conjecture that the elevated anomaly scores are due to the high temperature (generally 2°C higher than surroundings) and the steep temperature gradient in these parts of the powder bed.

VII. Conclusion

In this paper, we proposed an unsupervised deep learning approach for detecting potential anomalies in an AM system. As future work, we plan to apply the proposed technique to other industrial applications. We also plan to conduct a more in-depth theoretical analysis of the proposed technique.

References

[1] Syed A.M. Tofail, Elias P. Koumoulos, Amit Bandyopadhyay, Susmita Bose, Lisa O'Donoghue, and C.A. Charitidis. Additive manufacturing: Scientific and technological challenges, market uptake and opportunities. *Materials Today*, 21, 07 2017.

[2] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[3] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[4] François Chollet et al. Keras. https://keras.io, 2015.

[5] Katherine Heller, Krysta Svore, Angelos D Keromytis, and Salvatore Stolfo. One class support vector machines for detecting anomalous windows registry accesses. 2003.

[6] Yong Huang, Ming Leu, Jyoti Mazumder, and M Donmez. Additive manufacturing: Current state, future potential, gaps and needs, and recommendations. *Journal of Manufacturing Science and Engineering*, 137:014001, 02 2015.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167, 2015.

[8] Baihong Jin, Yuxin Chen, Dan Li, Kameshwar Poolla, and Alberto Sangiovanni-Vincentelli. A one-class support vector machine calibration method for time series change point detection. *arXiv preprint arXiv:1902.06361*, 2019.

[9] Baihong Jin, Dan Li, Seshadhri Srinivasan, See-Kiong Ng, Kameshwar Poolla, et al. Detecting and diagnosing incipient building faults using uncertainty information from deep neural networks. *arXiv preprint arXiv:1902.06366*, 2019.

[10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[11] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[12] Yuh-Jye Lee, Yi-Ren Yeh, and Yu-Chiang Frank Wang. Anomaly detection via online oversampling principal component analysis. *IEEE transactions on knowledge and data engineering*, 25(7):1460–1470, 2012.

[13] Dan Li, Guoqiang Hu, and CostasJ Spanos. A data-driven strategy for detection and diagnosis of building chiller faults using linear discriminant analysis. *Energy and Buildings*, 128:519–529, 2016.

[14] Dan Li, Yuxun Zhou, Guoqiang Hu, and Costas J Spanos. Fault detection and diagnosis for building cooling system with a tree-structured learning method. *Energy and Buildings*, 127:540–551, 2016.

[15] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148*, 2016.

[16] Jefferson Ryan Medel and Andreas Savakis. Anomaly detection in video using predictive convolutional long short-term memory networks. *arXiv preprint arXiv:1612.00390*, 2016.

[17] Tim Phillips. In-situ laser control method for polymer selective laser sintering (sls). Master's thesis, University of Texas at Austin, 2016.

[18] Clifton Phua, Vincent Lee, Kate Smith, and Ross Gayler. A comprehensive survey of data mining-based fraud detection research. *arXiv preprint arXiv:1009.6119*, 2010.

[19] Mayu Sakurada and Takehisa Yairi. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, page 4. ACM, 2014.

[20] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[22] Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using LSTMs. In *International conference on machine learning*, pages 843–852, 2015.

[23] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.