

# pTS-LSTM: Temperature prediction for fused filament fabrication using thermal image time series

Shenghan Guo<sup>a,\*</sup>, Rui Dai<sup>b</sup>, Haofan Sun<sup>b</sup>, Qiong Nian<sup>b</sup>

<sup>a</sup> The School of Manufacturing Systems and Networks, Arizona State University, 6075 Innovation Way West, Mesa, AZ 85212, USA

<sup>b</sup> The School for Engineering of Matter, Transport & Energy, Arizona State University, 501 E Tyler Mall, Tempe, AZ 85287, USA

## ARTICLE INFO

### Keywords:

In-situ thermal images

Time series

Long short-term memory

Fused filament fabrication

## ABSTRACT

In-situ thermography for Fuse Filament Fabrication (FFF) processes reveals the dynamic thermal behavior during printing. The data collected are thermal image time series. Their infrared (IR) intensity is visual evidence of heat-affected zone (HAZ) temperatures, which can be leveraged to train deep learning models, e.g., Long Short-Term Memory (LSTM), for real-time temperature prediction and process monitoring. Nonetheless, the data collection method and printing path may pose challenges for data modeling. Typically, the IR camera has a fixed position while the HAZ moves per the predetermined printing path. Consequently, the HAZ shifts in images and the features extracted from these thermal images show a “periodic” behavior over time. Such periodic patterns do not reflect any useful information about HAZ temperatures. Instead, they are noise hiding and interrupting the true temperature information, thus must be removed before using the data to train an LSTM model for temperature prediction. This study integrates a time series model, i.e., ARIMA, with Stacked LSTM to build a pTS-LSTM model that eliminates noisy patterns and predicts temperatures during FFF printing. The case study results show the outperformance of pTS-LSTM over conventional LSTM and classic Recurrent Neural Network models. pTS-LSTM is demonstrated to be promising for in-situ process monitoring with low-quality thermal images. In FFF practices, pTS-LSTM will be a preferred option over the commonly used deep learning models for thermal-image-based temperature prediction.

## 1. Introduction

Fused Filament Fabrication (FFF), also known as Fused Deposition Modeling (Stratasys trademark FDM™) [1], is an extrusion-based additive manufacturing (AM) process that deposits melted filament in a predetermined path to build parts layer by layer. Typically, the extruder of an FFF printer consists of a heating chamber and a nozzle. The heating chamber hosts the liquefier to melt the filament, allowing the molten material to exit from the small nozzle to form a thin bead of plastic that will adhere to the material it is laid on [2,3]. Due to the build mechanism, FFF fabricates durable parts with a high production rate and low expense [4,5], thus promising for both industrial and home uses [6]. On the other hand, the mechanism raises concerns about printing stability and quality. The temperature profiles of the contact region between the filament's hot end and the in-process layer, i.e., *extrudate deposition area* (or *heat-affected zone*, abbreviated as HAZ), are highly influential to the density and geometric accuracy of the printed parts. During FFF printing, unstable HAZ temperature profiles can cause abnormal thermal

expansion and shrink of the part, resulting in insufficient bounding, internal voids, and geometric deviation [3,7,8]. Hence, monitoring the HAZ and predicting its temperature profiles during printing is rather beneficial to FFF quality improvement.

State-of-the-art literature [9–11] has been using Infrared (IR) Thermography to monitor FFF processes in situ and provide timely feedback for the temperature profiles. In-situ thermography collected for HAZ reveals the dynamic thermal behavior during FFF printing. The data collected are thermal image time series. Their IR intensity is visual evidence for the temperature profiles, from which one can extract peak temperature, heating rate, cooling rate, etc. Fig. 1(a) shows the screenshot of a thermal video for a lab-based FFF process. Subsequent thermal image frames, as displayed in Fig. 1(b), show the transient HAZs, revealing the heat transfer status across adjacent part regions during printing. Deep learning (DL) models, e.g., Convolutional Neural Networks (CNNs) [12], can learn HAZ characteristics from these thermal images and predict the temperature profile or quality issues [13].

Despite the success in specific AM applications [13,14], current DL methods are limited for predictive analysis with FFF thermal videos due

\* Corresponding author.

E-mail address: [shenghan.guo@asu.edu](mailto:shenghan.guo@asu.edu) (S. Guo).

<https://doi.org/10.1016/j.jmapro.2023.09.053>

Received 25 June 2023; Received in revised form 24 August 2023; Accepted 26 September 2023

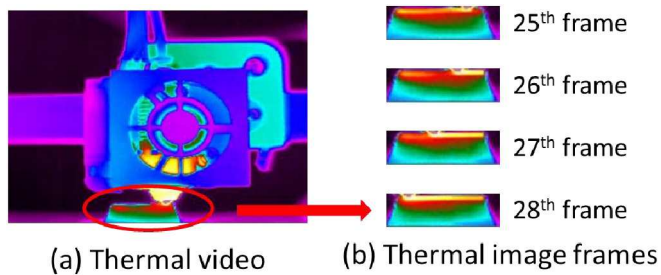
Available online 11 October 2023

1526-6125/© 2023 The Society of Manufacturing Engineers. Published by Elsevier Ltd. All rights reserved.

**Nomenclature**

|                                                  |                                                                        |
|--------------------------------------------------|------------------------------------------------------------------------|
| $X, x_t \in \mathbb{R}^n$                        | $n$ -dimensional time series extracted from videos                     |
| $l$                                              | Length of time series segments                                         |
| $\Phi, k; K$                                     | Kernel function; output of kernel function                             |
| $\lambda, \alpha, V$                             | Eigenvalue, eigenvector, and principal component in kernel PCA         |
| $X', x'_t \in \mathbb{R}^m$                      | $m$ -dimensional time series extracted from $X$ by kernel PCA          |
| $f, f_j$                                         | ARIMA model                                                            |
| $p, d, q; \mathcal{P}, \mathcal{D}, \mathcal{Q}$ | Orders in an ARIMA model $f$ ; parameter sets of $(p, d, q)$           |
| $L; \theta_i, \psi_i; \epsilon_t$                | Lag operator; coefficients; error term in ARIMA model                  |
| $E, e_t; h_t$                                    | Residual TS, input to an LSTM block; output of an LSTM block           |
| $C, I, O, F$                                     | Memory cell, input gate, output gate, and forget gate of an LSTM block |

|           |                                                                        |
|-----------|------------------------------------------------------------------------|
| $W, U, b$ | Weights of the input, recurrent connections, and bias in an LSTM block |
| AM        | Additive manufacturing                                                 |
| FFF       | Fused Filament Fabrication                                             |
| IR        | Infrared                                                               |
| HAZ       | Heat-affected zone                                                     |
| TS        | Time series                                                            |
| DL        | Deep Learning                                                          |
| CNN       | Convolutional Neural Network                                           |
| RNN       | Recurrent Neural Network                                               |
| LSTM      | Long Short-Term Memory                                                 |
| ARIMA     | Autoregressive Integrated Moving Average                               |
| PCA       | Principal Component Analysis                                           |
| AIC       | Akaike Information Criterion                                           |
| ROI       | Region of interest                                                     |
| MSE       | Mean Squared Error                                                     |
| CV        | Cross-Validation                                                       |



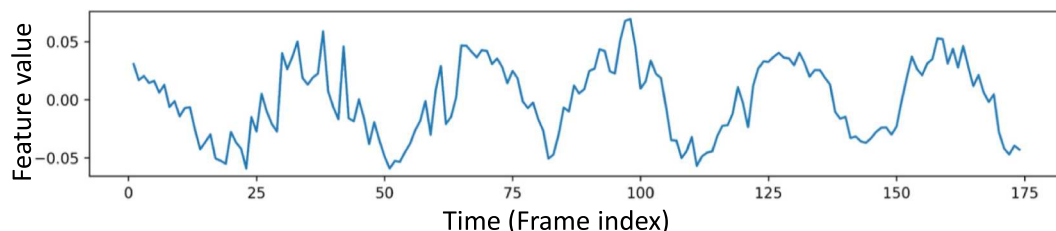
**Fig. 1.** Thermal video of a lab-based FFF process: (a) thermal video for an FFF process, (b) subsequent thermal image frames from the video.

to data-related issues, specifically *periodic patterns in thermal videos*. A thermal video is essentially an imaging time series. The frames capture an ongoing FFF process, thus reflecting the evolution of HAZ and heat transfer status from time to time. When the data collection method is ideal, e.g., having a sensor that captures high-resolution, complete images of the HAZ all the time, the thermal videos acquired would be free of excessive noise or missing information. A DL model can learn from these videos conveniently and predict HAZ temperatures with high accuracy [15]. Nonetheless, such high-quality thermal videos are hardly acquired from real FFF applications. The data collection can be restricted – the sensor is usually posed from a certain angle to the printing platform and captures the HAZ plus a large background (which is useless, peripheral information). The thermal images in Fig. 1(a) are an example of restricted data collection – huge background and imperfect sensor angle. Meanwhile, the HAZ would “shift” in subsequent thermal image frames following the printing path of FFF. The predefined printing path determines the back-and-forth nozzle movements in the camera view. Consequently, the shifts of HAZ show *periodicity*, which

can be visually identified from Fig. 1(b). With preliminary data processing and feature extraction, such periodicity is revealed as the typical “periodicity” in time series (TS) [16] (Fig. 2).

The periodic patterns in FFF thermal videos are irrelevant to the HAZ temperature profile, thus are noise and must be removed during DL-based prediction. Unfortunately, existing DL methods for FFF or other AM processes rarely address the data periodicity issue. They tend to assume high-quality, non-noisy thermal images. They are of limited use when the best thermal videos collectible are paradigms shown in Fig. 1. Targeting this issue, this study proposes a novel DL model, Long Short-Term Memory for periodic TS (abbreviated as pTS-LSTM), which integrates TS analysis, specifically Autoregressive Integrated Moving Average (ARIMA) [17,18], with Stacked LSTM [19] to enable automatic characterization and removal of periodicity in TS extracted from FFF thermal videos, and use the residual TS to predict HAZ temperatures. By removing the periodic pattern, pTS-LSTM preserves the relevant information for HAZ temperatures, avoiding the interruption from the data collection method and printing path.

This work will contribute to in-situ process monitoring of FFF from both methodological and practical perspectives. Hybrids of TS analysis and LSTM have been studied in non-manufacturing fields [20–23] but not yet explored for thermal image TS from AM. The benefits of such hybrid models, e.g., pattern characterization and forecasting, can be helpful for analyzing in-situ thermal images from FFF that are subject to noisy periodicity. pTS-LSTM integrates ARIMA and Stacked LSTM at the algorithm level, enabling real-time periodicity reduction from upcoming TS and LSTM-based HAZ temperature prediction. Unlike current DL methods that are only useful on high-quality thermal images/videos, pTS-LSTM is compatible with noisy data interrupted by data collection angle and printing path, thus allowing the adoption of DL in FFF applications with imperfect data collection capability. pTS-LSTM also has the potential to be generalized to other AM processes where thermal image TS are collected during printing.



**Fig. 2.** Features extracted from the thermal video of FFF processes form time series with periodicity. Feature extraction was done with kernel PCA.

The rest of this paper is organized as follows. Section 2 will review state-of-the-art literature for temperature monitoring in FFF and DL-based temperature prediction with thermal imaging data. Section 3 will elaborate on the technical details of pTS-LSTM development, followed by a case study in Section 4 to demonstrate pTS-LSTM on real data from a lab-based FFF process. Section 5 will conclude the paper and highlight future research directions.

## 2. Literature review

Real-time analysis of temperatures during FFF printing is enabled by two components: (a) in-situ sensing and processes monitoring and (b) predictive analysis of temperatures with thermal imaging data. This section reviews studies related to both and uncovers the research gaps in integrating TS models with LSTM for real-time prediction.

### 2.1. Temperature monitoring in FFF

Recent studies have made progress in temperature sensing and monitoring for FFF processes. Haro et al. [24] showed that temperature sensors could be used for FFF process monitoring. By applying temperature sensors, the researchers monitored in-situ temperature variation, including nozzle temperature and printing chamber temperature, to systematically study their linkage with filament diameter in use. Kousiatz and Karalekas [25] embedded Fiber Bragg Grating and Thermal Couple sensors into the samples at their 3rd layer and 20th layer, either longitudinally or transversely, to obtain in-situ loading profiles and TS temperature profiles. By comparing these two profiles, the researchers concluded that as long as an initial reference temperature value can be obtained and the temperature variation remains steady, the complete temperature profiles during the whole procedure can be obtained from the recorded Bragg peak wavelengths. Though the above studies demonstrated the possibility of sensor-based temperature monitoring in FFF, they have limited feasibility in practice due to the number and sensitivity restrictions of the sensors needed for collecting the temperature parameters at the positions relative to the nozzle.

Other researchers leveraged IR cameras to collect thermography containing the TS temperature field on the HAZ and ambient conditions [9,26,27]. Seppala and Migler [9] used IR thermography to observe the temperature evolution at the welding zone in the thermoplastic AM

process. Malekipour et al. [26] conducted thermal experiments to analyze parameters' effects on FFF printing quality and concluded that a fabrication process with more even temperature distribution would improve the mechanical properties of printed specimens. The issues with IR cameras are the accuracy and completeness of temperature profiles. The quality of data collected is susceptible to the IR camera position; the data provided a 2D profile, while the temperature evolution in FFF is a 3D process.

To simplify the temperature monitoring procedure, some researchers attempted to build physics models in combination with sensor data collection and analysis. For instance, Ravoori et al. [28] applied an analytical model to the FFF printing process, in which the energy conservation law and thermal distribution functions were applied to account for the heat transfer process during FFF, fitting well with their temperature data from the thermocouples. They concluded that, by manipulating the temperature parameters, desired filament-to-filament bonding could be achieved, leading to novel, spatially varying orthotropic parts. Similar efforts can also be found in Lu and Wang [29], Lu and Wang [30], which built a physics-based compressive sensing model to minimize the required number of sensors and the amount of data collected. A 1D heat transfer model was developed to manipulate the process parameters and optimize the bonding property between ABS layers.

The progress in sensor-based temperature monitoring for FFF enables further studies of the temperature profiles in FFF. To perform real-time analysis for HAZ temperatures and generate timely feedback, thermal image analysis and modeling are essential to inspect the sensor data from FFF and identify the printing quality [31].

### 2.2. Thermal-image-based temperature prediction in FFF

Machine Learning (ML) and DL models have been useful tools for automatic information extraction and analysis of thermal imaging data from AM processes [13,14,32–34]. There are recent works applying DL models on in-situ thermography of FFF for temperature or defect prediction, e.g., Saluja et al. [15], Jin et al. [35], Wang et al. [36]. However, thermal images collected from FFF processes may have a low quality that impedes effective learning by DL models. For example, the IR thermography in Prajapati et al. [27] observed the in-plane (x-y) view of the 1st layer but could not see the subsequent ones during printing because the previously deposited layers had absorbed the IR intensity and obscured the camera view. It was pointed out in Raplee et al. [37] that distinguishing and weakening the errors in IR measurement would significantly decrease the noisy radiation and facilitate thermal-image-based temperature monitoring. In FFF applications, the data collection procedure is difficult to improve due to physical restrictions, e.g., the sensor's distance, angle, and position relative to the nozzle. To enable temperature prediction in FFF with low-quality thermal imaging data, novel ML/DL models may help. Unfortunately, the existing literature on DL-based temperature or quality prediction for AM does not offer a solution. A novel DL model, specifically an LSTM, that learns the temporal evolution of HAZ temperature and meanwhile resists the irrelevant periodicity (i.e., noise) in thermal image TS is in imperative need, which is yet to be developed in this study.

### 2.3. Hybrids of ARIMA and LSTM

Improving LSTM's learning outcome and prediction performance on TS with noisy patterns has drawn discussion in recent literature. A promising solution is the hybrid use of ARIMA [17,18] and LSTM. As an advanced TS model, ARIMA integrates the basic TS models, i.e., Autoregressive and Moving Average [38], to represent the linear dependency parsimoniously and predict future TS values. ARIMA can identify the patterns or trends in TS and separate the patterns from transient information in a TS observation [17].

State-of-the-art literature proposed several ways of integrating

**Table 1**  
Pseudo code for pTS-LSTM training and implementation.

|                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------|
| <b>Training:</b>                                                                                                            |
| Obtain kernel PCA results, i.e., training data $X'$ ;                                                                       |
| Define the ARIMA parameter sets, $\mathcal{P}, \mathcal{D}, \mathcal{Q}$ ;                                                  |
| For $j$ in 1: $m$                                                                                                           |
| Find optimal ARIMA parameters $(p_j^*, d_j^*, q_j^*)$ ;                                                                     |
| Fit $f_j = \text{ARIMA}(p_j^*, d_j^*, q_j^*)$ ;                                                                             |
| Calculate residual $[e_{1j}, e_{2j}, \dots, e_{Tj}]^T$ ;                                                                    |
| End                                                                                                                         |
| Combine residuals from all the $m$ PCs to obtain $E$ ;                                                                      |
| Partition $E$ into TS segments of length $l$ with 1-step rolling forward, denoted by $E_1, E_2, \dots$ ;                    |
| Train Stacked LSTM with $E_1, E_2, \dots$ to obtain pTS-LSTM;                                                               |
| <b>Implementation:</b>                                                                                                      |
| Obtain the trained pTS-LSTM and $f_j, j = 1, \dots, m$ ;                                                                    |
| Accumulate new thermal images and apply kernel PCA;                                                                         |
| Obtain new data $X'_t$ and form TS segments with $l$ instances;                                                             |
| For $j$ in 1: $m$                                                                                                           |
| Use $f_j$ to forecast $\hat{x}'_{tj}$ ;                                                                                     |
| Calculate $e_{tj} = x'_{tj} - \hat{x}'_{tj}$ ;                                                                              |
| Append $(x'_{tj}, e_{tj})$ to all previous instances for the $j$ th PC;                                                     |
| Refit $f_j = \text{ARIMA}(p_j^*, d_j^*, q_j^*)$ for the $j$ th PC using $\{(x'_{1j}, e_{1j}), \dots, (x'_{tj}, e_{tj})\}$ ; |
| End                                                                                                                         |
| Combine residuals of the $m$ PCs from the new thermal images to form $E_t$ ;                                                |
| Feed $E_t$ into pTS-LSTM to predict HAZ temperature profile $y_{t+1}$ .                                                     |

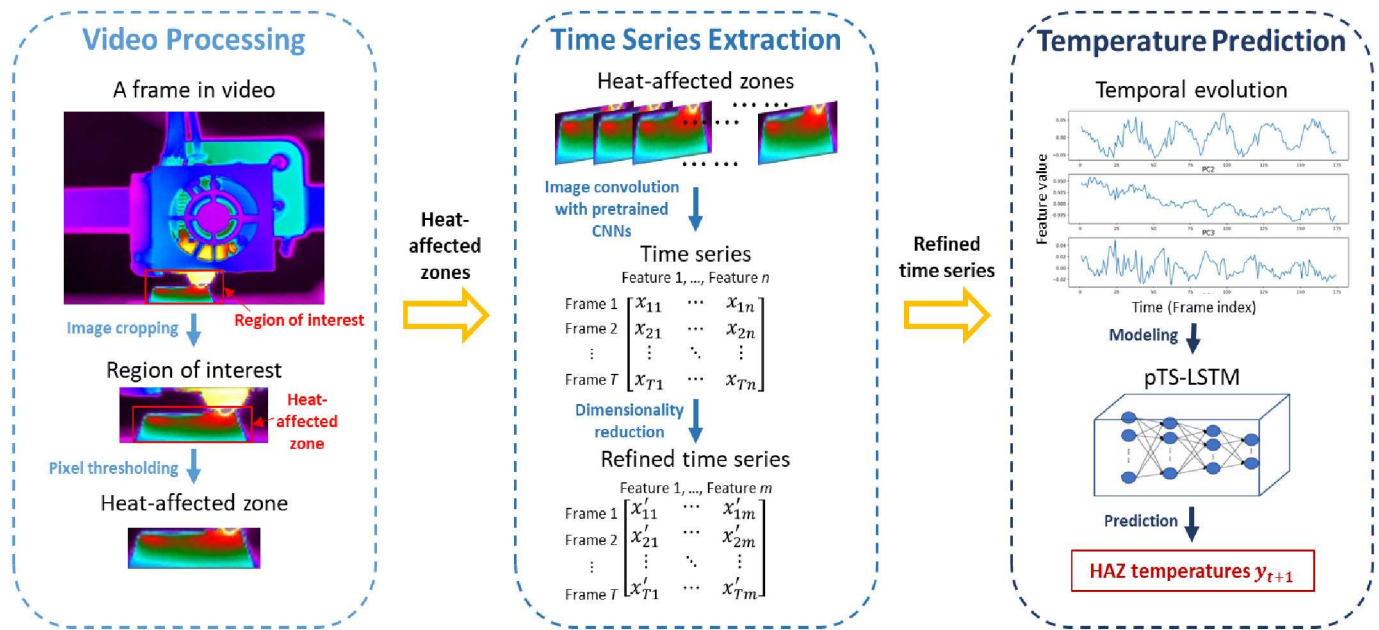


Fig. 3. Flowchart of using pTS-LSTM for temperature prediction in FFF.

ARIMA and LSTM. One major way is using ARIMA and LSTM as two subsequent steps for TS modeling or prediction [20–23]. Depending on the objective of the study, some works [20,23,39,40] used ARIMA first to model TS patterns or remove the noise and then fed the residuals to LSTM for prediction, while others [21,41–46] leveraged LSTM for error prediction and then reconstructed/forecast the TS data based on ARIMA. Parallel integration of ARIMA and LSTM was explored in Jin et al. [21] and Wu et al. [47], who used the outputs from ARIMA and LSTM as the input features of a statistical model for prediction. Indeed, the above works are inspiring and facilitated prediction with TS. But they used ARIMA and LSTM separately rather than integrating them into one algorithm for real-time implementation. Some studies [22,23,43] have the potential of real-time prediction, but they did not provide a generalizable algorithm, and the application field was not AM. Consequently, the existing hybrids of ARIMA and LSTM are limited in their applicability to in-situ thermal image TS from FFF or other AM processes.

To fill in the research gap, this study will integrate ARIMA and LSTM at the algorithm level to develop pTS-LSTM. In model training, ARIMA's optimal model parameters (see Section 3.2) will be identified as a separate step, and the fitted ARIMA will process the training TS to aid in pTS-LSTM training, specifically the training of a stacked LSTM model structure. In prediction, pTS-LSTM will be an integrated algorithm for ARIMA-enabled periodicity removal, ARIMA coefficient update, and LSTM prediction (see Section 3.4.2 and Table 1). In contrast to the existing hybrids of ARIMA and LSTM, pTS-LSTM will fit in in-situ temperature prediction for FFF and can be extended to other AM processes.

### 3. Method development

This study proposes pTS-LSTM for temperature prediction in FFF with low-quality thermal imaging data. The method is completed with three steps: (1) feature extraction from thermal videos (or equivalently, thermal image TS), (2) ARIMA modeling of the periodic pattern, and (3) development of pTS-LSTM model.

#### 3.1. Method overview

To build a pTS-LSTM for temperature prediction in FFF, several tasks need to be done, as shown in Fig. 3. First, thermal videos from FFF processes need to be acquired and processed. The HAZ in each frame is

identified and extracted. For objects of simple geometry, e.g., triangles, image thresholding [48] can do the job. The HAZs extracted from subsequent frames form an imaging TS. Image convolution with pretrained CNNs [49] extracts feature vectors from each image to form high-dimensional TS. Dimensionality reduction method, e.g., kernel PCA [50], can be used to refine the TS. The refined TS becomes the training data to build a pTS-LSTM model, or input to an established pTS-LSTM for real-time HAZ temperature prediction.

#### 3.2. Feature extraction from thermal image time series

Thermal videos are essentially TS of images. There are temporal connections across frames due to the continuous printing process in FFF. To develop an LSTM model structure for temperature prediction with the videos, optical features must be extracted from individual frames to form TS. This is done with pretrained CNNs.

##### 3.2.1. Feature extraction with pretrained CNNs

CNNs are designed to learn from images. In a CNN model, convolutional layers use filters, which are small receptive fields in the tensor form [51], to extract features from input images. Features from a convolutional layer form a tensor. A CNN takes the input images, extracts features with several convolutional layers, then flattens the feature matrix into a feature vector and maps it to the response (or “label”) with a couple of fully-connected layers of neurons [52]. The number of convolutional layers, as well as other layer types, can be determined by the user. There are a variety of commonly used CNN structures, e.g., VGG16 [53], ResNet [54], DenseNet [55]. These deep CNN models can be trained on rather large datasets, e.g., ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) data [56], and then applied to different datasets and solve new problems. Such models are referred to as pretrained CNNs [49].

Conventionally, CNNs are trained from scratch to do regression or classification tasks [57]. Pretrained CNNs, on the other hand, can be used directly for feature extraction from arbitrary images [58]. A pretrained CNN model for classification would take the image and map it to the probabilities that the object belongs to the existing classes, regardless of its actual object type. For feature extraction, the user does not take the final output but the feature vector from one of the fully-connected layers. A typical practice is to take the feature vector from



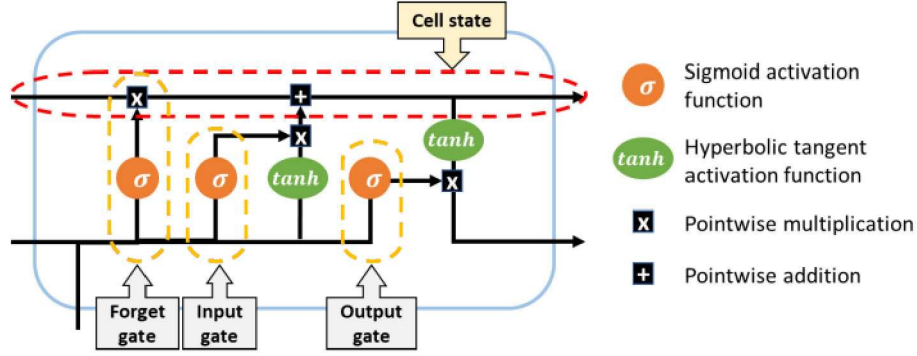


Fig. 4. LSTM memory cell topology adapted from Guo et al. [65].

the 1st fully connected layer [59]. The extracted feature vector is long, i. e., 4096 elements with a pretrained VGG16, 2048 elements with a pretrained ResNet50, and 1024 elements with a pretrained DenseNet121. Such a feature vector can be extracted from each frame in a thermal video. Feature vectors from subsequent frames form a high-dimensional TS.

### 3.2.2. Dimensionality reduction with kernel PCA

The features extracted by pretrained CNNs are high-dimensional TS. Kernel Principal Component Analysis (kernel PCA) can be leveraged to reduce the TS to its intrinsic dimension and meanwhile preserve the meaningful data properties [60]. Kernel PCA is an extension of the conventional PCA. While PCA uses orthogonal linear transform to project high-dimensional data to low-dimensional space, kernel PCA leverages kernel methods [61] to perform the transformation in a reproducing kernel Hilbert space. It is therefore superior to PCA in dimensionality reduction for nonlinear, complex data.

Denote the high-dimensional TS from a pretrained CNN by  $X = [x_1, x_2, \dots, x_T]^T$ ,  $x_t \in \mathbb{R}^n$  with  $n$  large,  $t = 1, \dots, T$ . A function,  $\Phi$ , maps  $x_t$  to a high-dimensional feature space, i. e.,  $x_t \rightarrow \Phi(x_t)$ . Given arbitrary feature vectors  $u$  and  $v$ , a  $T$ -by- $T$  kernel is created as [50,61],

$$K = k(u, v) = \Phi(u)^T \Phi(v) \quad (1)$$

A principal component (PC) from kernel PCA is

$$V = \sum_{t=1}^T \alpha_t \tilde{\Phi}(x_t), \tilde{\Phi}(x_t) = \Phi(x_t) - \frac{1}{T} \sum_{s=1}^T \Phi(x_s) \quad (2)$$

Vector  $\alpha = [\alpha_1, \dots, \alpha_T]$  is the eigenvector of  $K$ , whose length is chosen to have  $\|V\| = 1$  ( $\Leftrightarrow \|\alpha\|^2 = 1/\lambda$  for  $K$ 's eigenvalue  $\lambda$ ) [50]. In this study, the kernel is chosen to be the radial basis function [62]:

$$K = k(u, v) = \exp(-\gamma \|u - v\|^2) \quad (3)$$

where  $\gamma$  is a free parameter. Kernel PCA reduces the dimensionality of TS from  $n$  to  $m$ , which is the number of PCs preserved. Denote the reduced TS by  $X' = [x'_1, x'_2, \dots, x'_T]^T$ ,  $x'_t \in \mathbb{R}^m$ ,  $m \ll n$ .

### 3.3. ARIMA: modeling of periodic pattern in TS

Due to the periodic shifts of HAZ during FFF printing process, the TS extracted from thermal videos show periodic behavior. Such periodicity exists in  $X'$ . ARIMA [17,18] is a classic TS model that can be leveraged to characterize the periodic pattern in  $X'$ . It is parameterized by  $(p, d, q)$ , where  $p$  is the number of time lags in its autoregressive component,  $d$  is the degree of differencing, and  $q$  is the order of its moving-average component. An ARIMA( $p, d, q$ ) model without drift is [17]:

$$\left(1 - \sum_{i=1}^p \psi_i L^i\right) (1 - L)^d x'_{ij} = \left(1 + \sum_{i=1}^q \theta_i L^i\right) \epsilon_{ij},$$

$$j = 1, 2, \dots, m \quad (4)$$

where  $L$  is the lag operator,  $\psi_i$  are the coefficients for the autoregressive part,  $\theta_i$  are the coefficient for the moving average part, and  $\epsilon_t$  are error terms.

Assuming data stationarity [18], if model parameters  $(p, d, q)$  are specified, Eq. (4) can be fitted to  $X'$  to functionally characterize the periodic pattern. Note that each column (or equivalently, PC) in  $X'$  may have different periodicity, so Eq. (4) is preferred to be fitted to each column. ARIMA enables the removal of periodicity from TS by subtracting  $x'_t$  by the model-fitted value,  $\hat{x}'_t$ . The remaining part,  $e_t = x'_t - \hat{x}'_t$ , form a residual TS that can be input to pTS-LSTM (see Subsection 3.4.2 for ARIMA model fitting).

### 3.4. pTS-LSTM: temperature prediction with residual TS

When it comes to predictive analysis with TS data, LSTM and its variants [63] are effective and well-adopted options. pTS-LSTM is developed upon a Stacked LSTM model [19,64,65]. It integrates ARIMA models to characterize and forecast the periodicity for one step, preserving the residuals of TS as the input for temperature prediction with Stacked LSTM.

#### 3.4.1. LSTM

LSTM was developed from Recurrent Neural Networks (RNNs) to tackle the “vanishing gradient” problem. Specifically, when using gradient-based algorithms [66] to update the weights in conventional RNNs proportionally to the partial derivative of the error function per training iteration, the gradient can be vanishing and prevent the weights from updating [67]. LSTM incorporates gate units to allow for constant error flow through special, self-connected units, thus preventing “vanishing gradient” [63,68].

An LSTM block is formed with a self-connected memory cell ( $C$ ), an input gate ( $I$ ), an output gate ( $O$ ), and a forget gate ( $F$ ) (Fig. 4). For time  $t$ , the input for an LSTM block is  $e_t$  and the output (hidden state vector) from an LSTM block is  $h_t$ . The activation equations are [68,69]:

$$F_t = \sigma(W_F e_t + U_F h_{t-1} + b_F)$$

$$I_t = \sigma(W_I e_t + U_I h_{t-1} + b_I)$$

$$O_t = \sigma(W_O e_t + U_O h_{t-1} + b_O)$$

$$\tilde{C}_t = \sigma(W_C e_t + U_C h_{t-1} + b_C)$$

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t$$

$$h_t = O_t \odot \sigma(C_t) \quad (5)$$

where  $W$  are weights of the input,  $U$  are recurrent connections, and  $b$  are

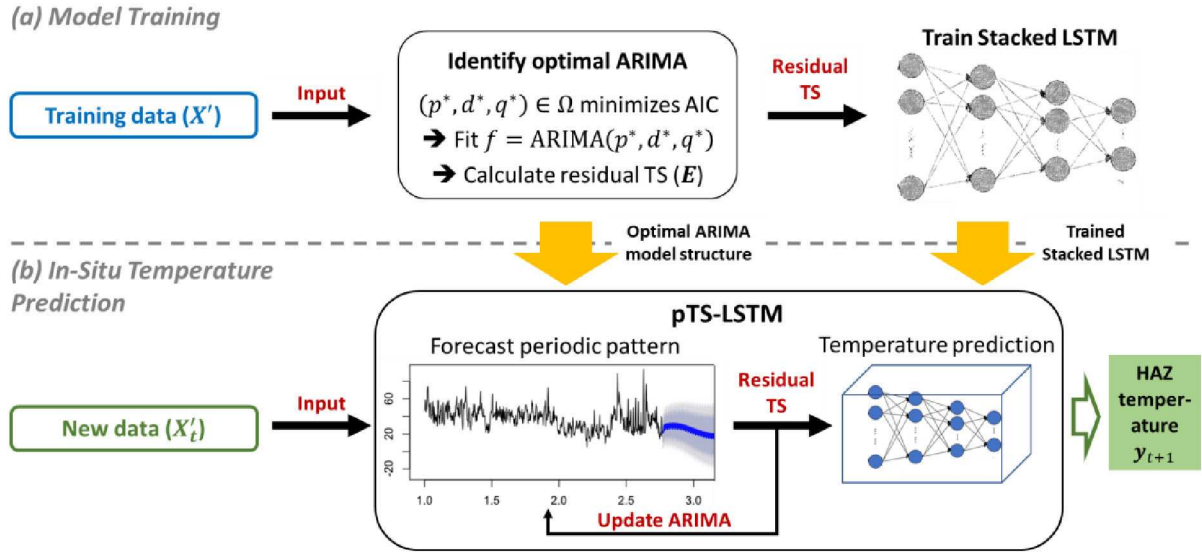


Fig. 5. pTS-LSTM: (a) model training, (b) in-situ temperature prediction with trained pTS-LSTM.

bias. Operator  $\circ$  is the Hadamard product (or element-wise product) [70].

The cell state is the “memory” of LSTM that carries relevant information throughout the processing of TS data. Even information from earlier time steps can be passed to later time steps, thus reducing the effects of “vanishing gradient”. An LSTM layer is formed with multiple cells, and several LSTM layers can be concatenated sequentially to form a Stacked LSTM, which has even better learning ability than simple LSTM models.

#### 3.4.2. pTS-LSTM

pTS-LSTM is developed by integrating a Stacked LSTM with an ARIMA model. For a new frame at time  $t$ , feature vector  $x_t$  would be extracted by a pretrained CNN and then converted to  $x'_t$  with kernel PCA. The ARIMA model fits the periodic pattern as  $\hat{x}_t$  and subtract it from  $x'_t$  to generate residual  $e_t$ .  $e_t$  is appended to residuals of previous  $l-1$  frames to form a TS segment of length  $l$ ,  $E_t = [e_{t-l+1}, e_{t-l+2}, \dots, e_t]^T$ , as the input for Stacked LSTM to predict HAZ temperatures,  $y_{t+1}$ , at time  $t+1$ .

Unlike CNNs, there are not many benchmark Stacked LSTM architectures. The model structure of Stacked LSTM needs to be adjusted per the training data. Fig. 6 shows the Stacked LSTM adopted in this study. It consists of 4 LSTM layers, with the layers containing 32, 16, 8, and 4 memory cells from left to right. Each LSTM layer adds one level of abstraction, and eventually map the input  $E_t$  to predicted temperatures  $y_{t+1}$ . An L2-regularized loss function [71],  $\mathcal{L}'$ , is adopted for the Stacked LSTM to avoid overfitting:

$$\mathcal{L}' = \mathcal{L} + \beta \|W\|^2 \quad (6)$$

where the original loss function  $\mathcal{L}$  is mean squared error (MSE),  $\beta$  is the hyperparameter to control the level of regularization and set to be  $10^{-5}$ . During model training,  $\mathcal{L}'$  superimposes the weights of each LSTM layer according to the L2 norm, as in Ridge regression [72], thus lowering the model complexity to avoid overfitting.

Another constitutive part of pTS-LSTM is ARIMA. Its optimal model structure is identified during model training of pTS-LSTM (Fig. 5(a)). Recall that  $m$  PCs are preserved from kernel PCA to form the TS. Each PC may show different periodic patterns, so an optimal ARIMA model exists for individual PC. For the  $j$ th PC,  $j = 1, 2, \dots, m$ , the model structure of ARIMA, as determined by its orders  $(p_j, d_j, q_j)$ , are found with a greedy search in user-defined parameter sets,  $\Omega := \mathcal{P} \times \mathcal{D} \times \mathcal{Q}$ .

$\forall (p_j, d_j, q_j) \in \Omega$ , an  $\text{ARIMA}(p_j, d_j, q_j)$  model would be fitted to the training data for the  $j$ th PC. The Akaike Information Criterion (AIC) [73] for this model is obtained as fitness measure. The final ARIMA model orders are  $(p_j^*, d_j^*, q_j^*)$  that minimize AIC for the training data. Let  $f_j = \text{ARIMA}(p_j^*, d_j^*, q_j^*)$  be the optimal ARIMA model fitted to the training data of  $j$ th PC. The fitted values from  $f_j, j = 1, 2, \dots, m$ , are subtracted from training data  $X'$  to generate residual TS  $E$ , which is the training data for the Stacked LSTM.

During in-situ implementation of pTS-LSTM (Fig. 5(b)), the ARIMA works as follows:

1. Obtain  $x'_{tj}$  from the thermal video frame at time  $t$ ;
2. Forecast  $\hat{x}'_{tj}$  with ARIMA as  $\hat{x}'_{tj} = f_{(t-1)j} \left( x'_{(t-1-p'_j)j}, \dots, x'_{(t-1-q'_j)j}; e_{(t-1-p'_j)j}, \dots, e_{(t-1-q'_j)j} \right)$ ;
3. Calculate  $e_{tj} = x'_{tj} - \hat{x}'_{tj}$ ;
4. Append  $(x'_{tj}, e_{tj})$  to all previous instances (including the training data) for the  $j$ th PC;
5. Refit  $f_j = \text{ARIMA}(p_j^*, d_j^*, q_j^*)$  for the  $j$ th PC using  $\{(x'_{1j}, e_{1j}), \dots, (x'_{tj}, e_{tj})\}$ , which will forecast  $\hat{x}'_{(t+1)j}$ .

Step 5 refits the ARIMA model every time a new frame becomes available, which enhances the model fitness, or equivalently, the accuracy of characterizing the periodic pattern, at the expense of computing speed. The prediction time would be extended by  $<1$  min. It can be omitted if fast prediction is of priority. In that case, the ARIMA models fitted with training data will be used as they are throughout in-situ temperature prediction. The pseudo code for the entire training and implementation procedure for pTS-LSTM is given in Table 1.

Note that pTS-LSTM is applicable to HAZ temperature prediction for FFF-ed parts with varying sizes, printing speeds, and shapes. As a data-driven model, pTS-LSTM relies on good model training. If thermal image TS can be collected, processed, and train pTS-LSTM following the above methodology steps, then the trained pTS-LSTM model can predict HAZ temperature profiles for new parts despite their different size and build parameters. When extended to new FFF-ed parts, the user will need to adjust the Stacked LSTM model structure and training epochs based on

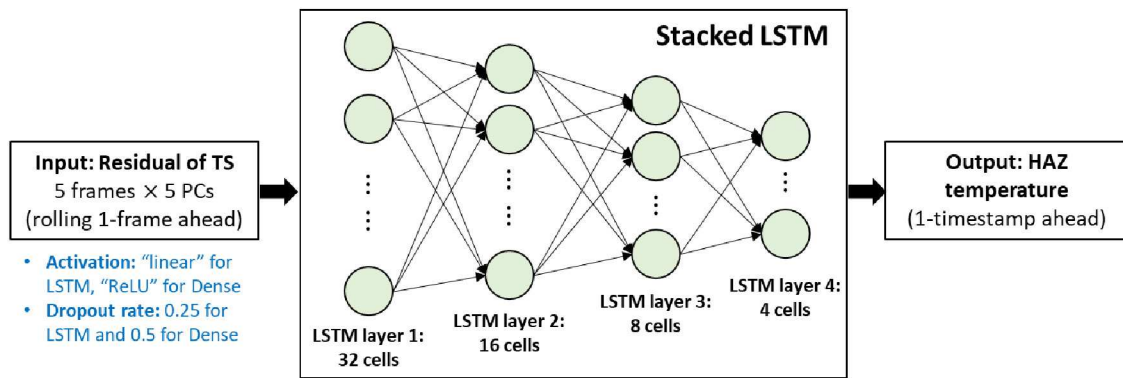


Fig. 6. The Stacked LSTM structure used in the current pTS-LSTM. The input and output layers are not visualized.

their data and computing facilities. The case study here (see Section 4) provides a reference pTS-LSTM model structure and training procedure and can be the benchmark for pTS-LSTM utilization on new parts.

#### 4. Case study

The proposed pTS-LSTM is implemented on a thermal video for a lab-based FFF process. A performance comparison is done between conventional LSTM models, pTS-LSTM, and classic RNNs.

##### 4.1. Data collection

Fig. 7 shows the experiment setup for specimen printing and data collection. A commercial FFF printer, Ender 3 Pro, was utilized. A triangular specimen with a 3 cm edge length in PLA (white) was printed to support real-time thermal image collection. The specimen had a 100 % infill grid pattern and around 2 mm in the depth direction (parallel to the camera direction) to minimize the nozzle path away from the camera. The layer thickness was set to 0.2 mm to ensure good printing quality. The printing speed was 45 mm/min.

A FLIRTM A320 30HZ thermal camera (Focal Plane Array (FPA), uncooled microbolometer) was manually focused on the sample horizontally at a 30 cm distance for in-situ monitoring of the FFF printing process. FLIRTM A320 camera is a non-contact thermal device operating in the spectral range of 7.5 to 13  $\mu$ m. The camera has an instantaneous field of view (FOV) of 25° and provides 16-bit images at an accuracy of  $\pm 0.2$  °C or  $\pm 2$  %. The time series imaging collected by the camera formed a thermal video. It captured 175 frames for the  $\sim 150$  layers of the FFF printing process. Each frame has a resolution of  $320 \times 240$  pixels.

To train DL models, the video frames need to be “labeled”. During in-situ temperature monitoring, the FLIR software recorded the point temperature of HAZs, the region right below the nozzle on the top layer

of the printing specimen and exported as the “labels”. Temperatures obtained from the thermal infrared data are correlated with surface emissivity. A default emissivity, 0.92, was applied in this case study. These labels were manually linked to each video frame to build a supervised dataset for DL model training.

##### 4.2. Data preprocessing

The raw thermal images collected by the FLIR camera captured a large background (refer to Fig. 1(a)), which was peripheral information that might compromise the subsequent feature extraction outcomes. Hence, data preprocessing was done to remove the large background in each video frame and prepare the dataset for DL model training and validation.

The data preprocessing was completed in two steps. First, the specimen was mainly in the lower part of each frame, corresponding to a fixed region in the image, i.e., 280th to 320th pixel from top to bottom and 150th to 250th pixel from left to right. Define this region as the *region of interest* (ROI), essentially the image segment under the nozzle head containing the HAZ and the entire layer being deposited. It was a fixed area cropped out of each raw thermal image. The cropping size here was selected per the raw image size and HAZ tracks. Next, each ROI segment was thresholded with its own median pixel values. A rectangular matrix of pixels was identified in each ROI as the HAZ. These HAZs formed the actual thermal image TS and then were used as the input for feature extraction (refer to Section 3.2).

##### 4.3. Results and discussion

The novelty of pTS-LSTM is the integration of ARIMA and Stacked LSTM for characterization and removal of data periodicity. This model design makes pTS-LSTM superior to conventional LSTM models in predicting HAZ temperatures for FFF processes. For demonstration, pTS-

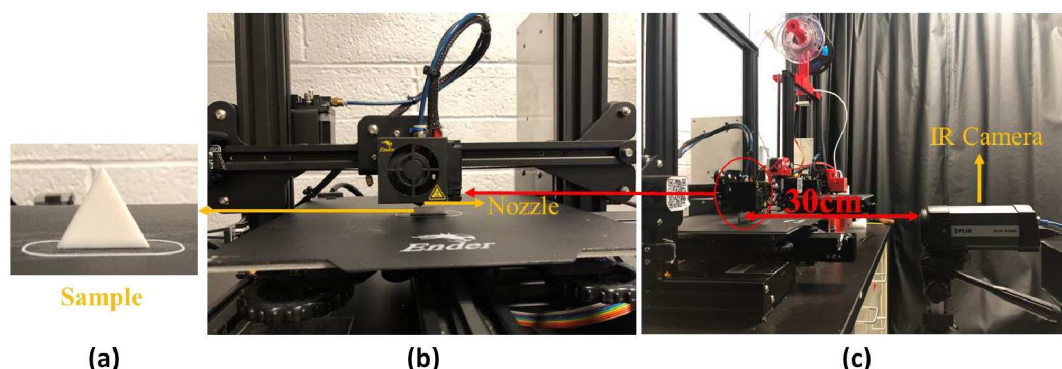
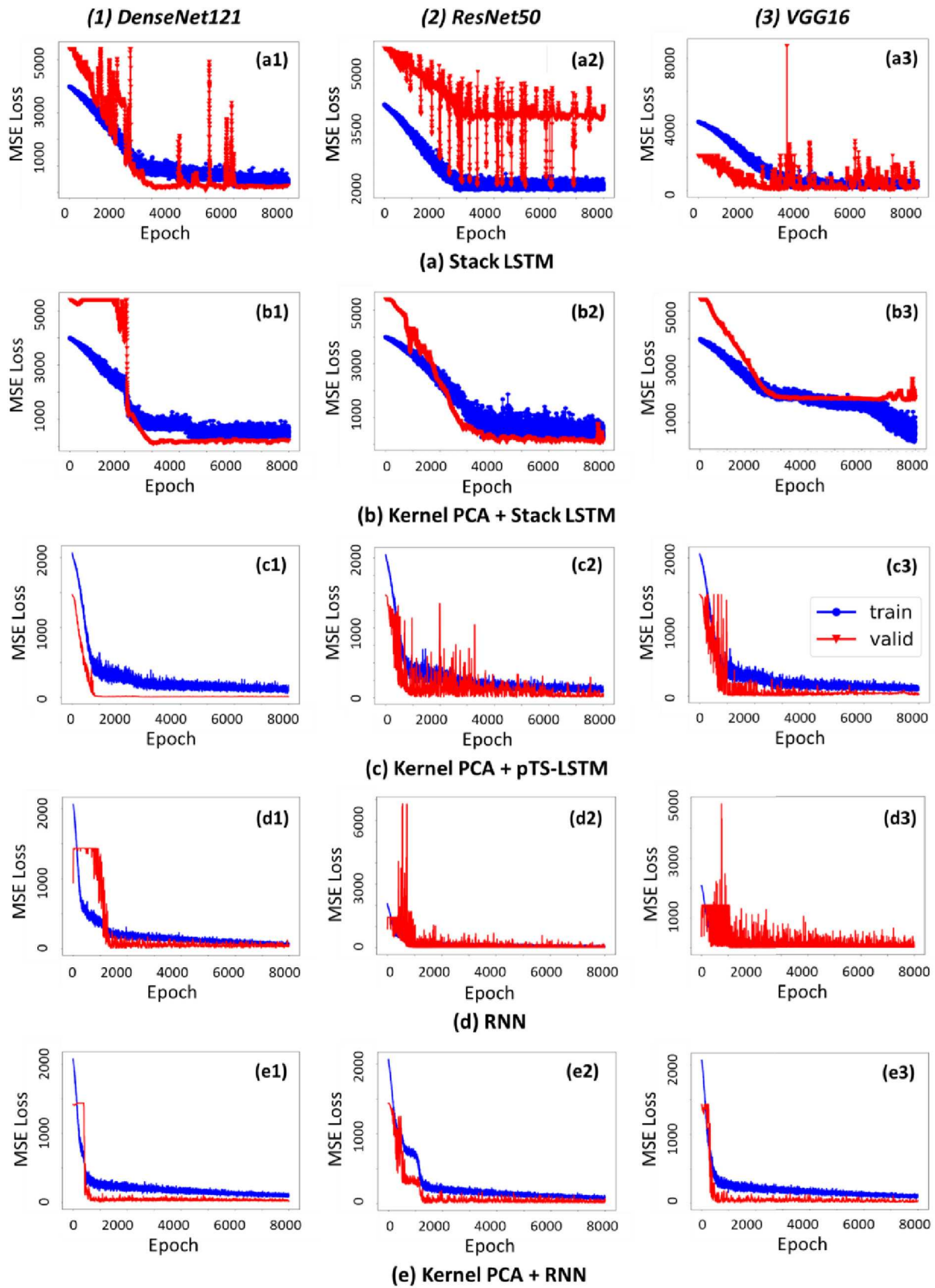


Fig. 7. Experiment setup for data collection: (a) printed sample, (b) front view of the printer, (c) side view of the printer and the IR camera.



**Fig. 8.** Training performance: (a1–3) Stacked LSTM, (b1–3) kernel PCA + Stacked LSTM, (c1–3) kernel PCA + pTS-LSTM, (d1–3) RNN, (e1–3) kernel PCA + RNN.



**Table 2**

Model training time in seconds based on V100 Nvidia GPU.

| Model       | Time(s)      |                           |                       |     |                  |
|-------------|--------------|---------------------------|-----------------------|-----|------------------|
|             | Stacked LSTM | Kernel PCA + Stacked LSTM | Kernel PCA + pTS-LSTM | RNN | Kernel PCA + RNN |
| DenseNet121 | 1077         | 830                       | 832                   | 720 | 560              |
| ResNet50    | 1210         | 811                       | 849                   | 681 | 522              |
| VGG16       | 1400         | 975                       | 1002                  | 893 | 711              |

LSTM was compared with conventional Stacked LSTM. TS extraction was done with pretrained CNNs, i.e., DenseNet121, ResNet50, and VGG16. Five models were compared: (a) Stacked LSTM, (b) kernel PCA + Stacked LSTM, (c) kernel PCA + pTS-LSTM, (d) RNN (2 RNN layers, the 1st layer contains 16 neurons and the 2nd layer contains 8 neurons), (e) kernel PCA + RNN (same RNN structure).  $m = 5$  PCs were preserved in kernel PCA. (a)–(c) was a fair comparison among conventional LSTM and the proposed pTS-LSTM with the Stacked LSTM structure in Fig. 6, which demonstrates how pTS-LSTM improves the model performance by reducing the data periodicity. (d) and (e) provided an additional comparison between pTS-LSTM and commonly used, shallow RNNs. For pTS-LSTM, the optimal ARIMA parameters were searched from sets  $\mathcal{P} = \{1, 2, 3, 4, 5, 6\}$ ,  $\mathcal{D} = \{0, 1\}$ ,  $\mathcal{Q} = \{1, 2, 3, 4, 5, 6\}$  as part of model training. All the  $(p, d, q)$  combinations formed from  $\mathcal{P}$ ,  $\mathcal{D}$ , and  $\mathcal{Q}$  were used to fit ARIMA models. The optimal parameter set gave the ARIMA model that achieved the smallest AIC among all combinations. These relatively small parameter sets would expedite pTS-LSTM model training at the expense of obtaining suboptimal parameter values, which is acceptable for the case study and can be further improved in real practice. For all (a)–(e), 5-fold cross-validation (CV) [74,75] is adopted to validate the method's accuracy and robustness (CV for TS can be found in Appendix A.1). In each replicate of CV, 1 fold is preserved as the testing data (37 frames), and the rest become the training data (148 frames), within which 10 % are used for training-stage validation. All the models are trained with 8000 epochs without batches. The TS segment length is  $l = 5$ .

#### 4.3.1. Training performance

Fig. 8 displays the training performance of models (a)–(e) with varying feature extraction methods. Within (a)–(c), the proposed model (c) obviously outperformed (a) and (b) – pTS-LSTM converged faster and to a nearly zero MSE loss. In contrast, (a) had overfitting issues throughout model training, and (b) did not fully converge, especially when VGG16 was used for feature extraction, though 8000 epochs had been completed. For (d) and (e), the model convergence was fast, which is likely due to the simpler model structure. However, overfitting issues were observed for (d)–(e) even after model convergence. With the adoption of kernel PCA in (e), the training performance was improved from (d) and the steadiest after convergence among (a)–(e).

**Table 3**

Average MSE (and standard deviation) in prediction across 5-fold CV.

| Pretrained CNN | MSE    | Stacked LSTM          | Kernel PCA + Stacked LSTM | Kernel PCA + pTS-LSTM      | RNN                        | Kernel PCA + RNN             |
|----------------|--------|-----------------------|---------------------------|----------------------------|----------------------------|------------------------------|
| DenseNet121    | Min    | 267.3325 (252.4384)   | 495.7927 (727.7109)       | 24.6847 (30.0127)          | <b>12.7631 (7.2039)</b>    | 14.5708 (13.5650)            |
|                | Mean   | 534.6227 (451.5825)   | 760.6500 (699.7481)       | <b>74.2074 (68.4914)</b>   | 81.3530 ( <b>40.1182</b> ) | 90.5040 (82.1456)            |
|                | Median | 557.5411 (595.5952)   | 632.8164 (766.2288)       | 67.4238 (88.9621)          | <b>59.5286 (34.7076)</b>   | 65.6630 (54.5501)            |
|                | Max    | 812.7719 (565.1169)   | 1365.8784 (746.3702)      | <b>130.5137 (108.4623)</b> | 277.7966 (181.3297)        | 284.5965 (287.2334)          |
| ResNet50       | Min    | 666.2767 (892.1636)   | 40.8302 (45.1437)         | 13.0784 (5.9971)           | <b>9.7916 (5.1778)</b>     | 11.8529 (6.4217)             |
|                | Mean   | 1104.0645 (1378.8525) | 364.5163 (337.7676)       | <b>33.9749 (41.9028)</b>   | 82.0963 (92.7912)          | 63.7859 ( <b>38.7440</b> )   |
|                | Median | 1105.0940 (1453.8470) | 278.1954 (409.4122)       | <b>16.4049 (6.9352)</b>    | 52.9101 (49.1507)          | 53.3044 (25.8283)            |
|                | Max    | 1608.9044 (1832.2020) | 968.2463 (854.7760)       | <b>72.4413 (123.8482)</b>  | 208.5660 (247.3396)        | 177.1793 (140.7060)          |
| VGG16          | Min    | 187.2257 (307.0892)   | 200.8680 (210.8844)       | <b>12.9007 (4.3450)</b>    | 57.9908 (85.6509)          | 40.9340 (43.7871)            |
|                | Mean   | 340.1107 (310.4982)   | 579.3239 (411.1842)       | <b>94.6323 (146.7450)</b>  | 159.8329 (187.4225)        | 195.3294 (175.5987)          |
|                | Median | 285.9179 (344.9747)   | 487.2709 (519.8764)       | <b>56.7453 (91.9067)</b>   | 153.1364 (189.0401)        | 194.2592 (199.2445)          |
|                | Max    | 636.5734 (458.6399)   | 1217.1738 (731.7707)      | <b>214.5510 (348.3276)</b> | 333.3135 (339.7511)        | 388.2570 ( <b>280.7160</b> ) |

Table 2 provides the training times for models (a)–(e) measured in seconds (based on V100 Nvidia GPU). Comparing all the models, the computing time was mainly affected by the feature dimensionality and model structures. (a) consumed the longest training among all models for all three feature extraction methods. When kernel PCA was used, the input feature dimensionality was much reduced. Consequently, (b) and (c) had shorter training time than (a). The same time-saving effect of kernel PCA was observed by comparing (d) and (e). For LSTM and RNN model structures, the RNN model structures used less training time. An additional factor for model training time was the feature extraction method. Compared with DenseNet121 and ResNet50, VGG16 features required a longer time to train the same model structure. For the proposed model (c), the model training times were at an intermediate level among all the models and close to those of (b). The trivial training time difference between (b) and (c) indicates that pTS-LSTM improved the model training performance (see Fig. 8) without increasing the computing burden.

#### 4.3.2. Prediction performance

The trained models (a)–(e) were used for temperature prediction in the lab-based FFF process. Table 3 shows the prediction performance, measured by accuracy (or average MSE) and robustness (or standard deviation of MSE) across 5-fold CV. Both performance metrics are the smaller, the better.

In Table 3, the smallest metrics have been highlighted in bold, which were mostly achieved by (c). When pretrained DenseNet121 was adopted for feature extraction, (d) had acceptable prediction performance – the average MSEs were relatively low, and their standard deviations were small. DenseNets are CNN model structures designed to overcome the “vanishing gradient” problem [55,76], which is confronted by RNNs as well. This capability of DenseNet121 led to better feature extraction results – the features well preserved the useful information about HAZ temperature profiles, thus improving RNN learning outcomes. For all other combinations of feature extraction methods and benchmark models, (c) had more favorable prediction performance than the benchmarks. Specifically, when pretrained ResNet50 was used, (c) achieved the best performance. It is noteworthy that some results of benchmark models revealed the “downside” of using kernel PCA – it facilitated model training but caused information loss and compromised the prediction accuracy. Comparing (a) and (b), as well as (d) and (e), one may observe higher MSE when kernel PCA was adopted for the same LSTM or RNN model.

#### 4.3.3. Summary

The above results illustrate the pTS-LSTM's superiority in handling thermal image TS. For a fixed Stacked LSTM structure, pTS-LSTM significantly improved both the training and prediction performance of conventional LSTM models. In model training, pTS-LSTM brought the MSE loss at the converged level from >1000 (for (a) and (b)) to nearly 0 and reduced overfitting; in prediction, the min, mean, median, and

max MSE of pTS-LSTM were respectively 93.97 %, 85.07 %, 88.86 %, 81.92 % lower than those of (a) and 85.52 %, 88.20 %, 90.60 %, 88.44 % lower than those of (b) (averaged for the three pretrained CNNs). When compared with classic RNNs, pTS-LSTM consistently performed well in both training and prediction, while shallow RNNs were relatively easy to train but might not predict HAZ temperature profiles accurately. In prediction, the mean, median, and max MSE of pTS-LSTM were respectively 36.06 %, 39.56 %, 51.31 % lower than those of (d) and 38.77 %, 45.78 %, 52.67 % lower than those of (e) (averaged for the three pretrained CNNs). Therefore, pTS-LSTM is a better option than the existing DL methods for modeling and predicting HAZ temperature profiles in FFF.

## 5. Conclusion and future work

This study developed a novel DL model, pTS-LSTM, for in-situ HAZ temperature prediction in FFF. pTS-LSTM integrated TS forecasting and Stacked LSTM in its model design, thus enabling automatic characterization and removal of noisy, irrelevant periodicity in data. This method targeted the data-level challenges brought by the data collection and printing path in FFF. It will benefit FFF and other AM processes where thermal images are collected as the major evidence of process monitoring and prognostics. The proposed method can be readily extended to other laser-based AM applications for temperature or quality prediction during the printing process of simple geometry.

In the future, several research directions can be pursued. First, pTS-LSTM can be extended for AM printing of complex geometry. Currently, pTS-LSTM handles data periodicity due to a simple printing path. As the object geometry gets sophisticated, the printing path will be unique, resulting in more complex patterns in thermal video data. Enhancing the

skills of pTS-LSTM for such complex data is an intriguing topic for future studies. Deeper LSTM structures and other RNN variants may be integrated into pTS-LSTM to handle the data complexity and noise reduction task. Second, kernel PCA was adopted here to reduce the dimensionality of CNN-extracted features. It facilitated model training by lessening the computing burden but unavoidably led to information loss. Using end-to-end 3D CNNs [77,78] for a 1-step extraction of spatial-temporal features from thermal video segments will be explored in the future extension of this work. Third, more advanced HAZ sensing and tracking will be explored. This issue is related to the geometry complexity of the printed object. For complex geometry, the HAZ location in subsequent frames will be subject to larger uncertainty. Advanced sensing, e.g., in-situ X-ray computer tomography, will be considered to capture high-quality HAZ temperature profiles for pTS-LSTM model training. More effective HAZ segmentation methods (compared with thermal image thresholding) will be developed to assist pTS-LSTM training and implementation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgement

We acknowledge the use of ASU Research Computing for data processing and model training. This study is partially supported by Arizona State University startup funds, NSF grant 1826439 and NSF grant 1762792.

## Appendix A

### A.1. *K*-fold cross-validation for time series

This study adopts *K*-fold cross-validation (CV), with  $K = 5$ , in model training and testing. The thermal imaging data are time series (TS), so the data split procedure in CV is slightly different from the conventional *K*-fold CV.

For TS,  $S = \{y_t\}, t = 1, \dots, n$ , subsequent observations  $y_t$  and  $y_{t+1}$  have temporal connections (or correlations) that must be preserved, so the order of instances in  $S$  should not be altered. In data split,  $S$  is first divided into 2 parts, with the first part being the initial training set, denoted by  $S_0$  and  $S_1$ . In this study,  $S_0$  contains the first 40 % instances in  $S$ . Next,  $S_1$  is divided into  $K$  equal-sized folds, denoted by  $s_1, s_2, \dots, s_K$ . The first  $K - 1$  folds have  $\lfloor |S_1|/K \rfloor$  instances, while the  $K$ th fold may have several more instances if  $|S_1|/K$  is not an integer.

During CV, the 1st iteration uses  $S_0$  for model training and  $s_1$  for model testing. Next,  $s_1$  is appended to  $S_0$  as the training set for the 2nd iteration, and  $s_2$  is used for model testing. Moving forward,  $s_k, k = 2, \dots, K - 1$  is appended to the training set from the previous iteration and  $s_{k+1}$  is used for model testing in the current iteration, until  $K$  iterations are completed. The performance metrics of model testing are the sample mean (or average) and standard deviation of the MSE losses across the  $K$  iterations, i.e.,  $\bar{\mathcal{L}} = \sum_{k=1}^K \mathcal{L}_k / K$  and  $\hat{\sigma}(\mathcal{L}) = \sqrt{\sum_{k=1}^K (\mathcal{L}_k - \bar{\mathcal{L}})^2 / (K - 1)}$ , respectively. The above procedure is elaborated in Hyndman and Athanasopoulos [74].

## References

- [1] Tao Y, Kong F, Li Z, Zhang J, Zhao X, Yin Q, et al. A review on voids of 3D printed parts by fused filament fabrication. *J Mater Res Technol* 2021;15:4860–79. <https://doi.org/10.1016/j.jmrt.2021.10.108>.
- [2] Volpato N, Kretschek D, Foggia JA, Gomez da Silva Cruz CM. Experimental analysis of an extrusion system for additive manufacturing based on polymer pellets. *Int. J. Adv. Manuf. Technol.* 2015;81(9):1519–31. <https://doi.org/10.1007/s00170-015-7300-2>.
- [3] Singh S, Singh G, Prakash C, Ramakrishna S. Current status and future directions of fused filament fabrication. *J. Manuf. Processes* 2020;55:288–306. <https://doi.org/10.1016/j.jmapro.2020.04.049>.
- [4] Popescu D, Zapciu A, Amza C, Baciu F, Marinescu R. FDM process parameters influence over the mechanical properties of polymer specimens: a review. *Polym. Test.* 2018;69:157–66.
- [5] Mazzanti V, Malagutti L, Mollica F. FDM 3D printing of polymers containing natural fillers: a review of their mechanical properties. *Polymers* 2019;11(7):1094.
- [6] Ransikarbum K, Ha S, Ma J, Kim N. Multi-objective optimization analysis for part-to-printer assignment in a network of 3D fused deposition modeling. *J. Manuf. Syst.* 2017;43:35–46. <https://doi.org/10.1016/j.jmsy.2017.02.012>.
- [7] Penumakala PK, Santo J, Thomas A. A critical review on the fused deposition modeling of thermoplastic polymer composites. *Compos Part B Eng* 2020;201:108336. <https://doi.org/10.1016/j.compositesb.2020.108336>.
- [8] Vanaei HR, Khelladi S, Deligant M, Shirinbayan M, Tcharkhtchi A. Numerical prediction for temperature profile of parts manufactured using fused filament fabrication. *J. Manuf. Process.* 2022;76:548–58. <https://doi.org/10.1016/j.jmapro.2022.02.042>.
- [9] Seppala JE, Migler KD. Infrared thermography of welding zones produced by polymer extrusion additive manufacturing. *Addit Manuf* 2016;12:71–6.
- [10] He K, Zhang Q, Hong Y. Profile monitoring based quality control method for fused deposition modeling process. *J. Intell Manuf* 2019;30(2):947–58. <https://doi.org/10.1007/s10845-018-1424-9>.
- [11] Huang T, Wang S, Yang S, Dai W. Statistical process monitoring in a specified period for the image data of fused deposition modeling parts with consistent layers. *J. Intell Manuf* 2021;32(8):2181–96. <https://doi.org/10.1007/s10845-020-01628-4>.

- [12] Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. In: 2017 International Conference on Engineering and Technology (ICET); 2017. <https://doi.org/10.1109/ICETechnol.2017.8308186>.
- [13] Tian Q, Guo S, Melder E, Bian L, Guo WG. Deep learning-based data fusion method for in situ porosity detection in laser-based additive manufacturing. *J. Manuf. Sci. Eng.* 2020;143(4). <https://doi.org/10.1115/1.4048957> (7/16/2022).
- [14] Guo WG, Tian Q, Guo S, Guo Y. A physics-driven deep learning model for process-porosity causal relationship and porosity prediction with interpretability in laser metal deposition. *CIRP Annals* 2020;69(1):205–8. <https://doi.org/10.1016/j.cirp.2020.04.049>.
- [15] Saluja A, Xie J, Fayazbakhsh K. A closed-loop in-process warping detection system for fused filament fabrication using convolutional neural networks. *J. Manuf. Process.* 2020;58:407–15. <https://doi.org/10.1016/j.jmapro.2020.08.036>.
- [16] Elfekey MG, Aref WG, Elmagarmid AK. Periodicity detection in time series databases. *IEEE Trans Knowl Data Eng* 2005;17(7):875–87. <https://doi.org/10.1109/TKDE.2005.114>.
- [17] Shumway RH, Stoffer DS. ARIMA Models. In: Shumway RH, Stoffer DS, editors. *Time Series Analysis and Its Applications: With R Examples*. Cham: Springer International Publishing; 2017. p. 75–163.
- [18] Wadi S, Almasarweh M, Alsarairah AA, Aqaba J. Predicting closed price time series data using ARIMA model. *Modern Appl. Sci.* 2018;12(11):181–5.
- [19] Heryadi, Y. and Warnars, H.L.H.S. "Learning temporal representation of transaction amount for fraudulent transaction recognition using CNN, Stacked LSTM, and CNN-LSTM". In: 2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom). 2017. <https://doi.org/10.1109/CYBERNETICSCOM.2017.8311689>.
- [20] Ashok A, Prathibhamol CP. Improved analysis of stock market prediction: (ARIMA-LSTM-SMP). In: 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE); 2021. <https://doi.org/10.1109/ICNTE51185.2021.9487745>.
- [21] Jin Y, Wang R, Zhuang X, Wang K, Wang H, Wang C, et al. Prediction of COVID-19 data using an ARIMA-LSTM hybrid forecast model. *Mathematics* 2022;10(21):4001.
- [22] Ji L, Zou Y, He K, Zhu B. Carbon futures price forecasting based with ARIMA-CNN-LSTM model. *Procedia Comput. Sci.* 2019;162:33–8. <https://doi.org/10.1016/j.procs.2019.11.254>.
- [23] Manowska A, Rybak A, Dylong A, Pielot J. Forecasting of natural gas consumption in Poland based on ARIMA-LSTM hybrid model. *Energies* 2021;14(24):8597.
- [24] Haro FB, de Agustín del Burgo JM, D'Amato R, Islán M, Heras ES, Alonso JMG, et al. Monitoring an analysis of perturbations in fusion deposition modelling (FDM) processes for the use of biomaterials. *J Med Syst* 2019;43(5):1–8.
- [25] Kousiatza C, Karalekas D. In-situ monitoring of strain and temperature distributions during fused deposition modeling process. *Mater. Des.* 2016;97:400–6.
- [26] Malekipour E, Attouy S, El-Mounayri H. Investigation of layer based thermal behavior in fused deposition modeling process by infrared thermography. *Procedia Manuf.* 2018;26:1014–22.
- [27] Prajapati H, Salvi SS, Ravoori D, Jain A. Measurement of the in-plane temperature field on the build plate during polymer extrusion additive manufacturing using infrared thermometry. *Polym. Test.* 2020;92:106866.
- [28] Ravoori D, Lowery C, Prajapati H, Jain A. Experimental and theoretical investigation of heat transfer in platform bed during polymer extrusion based additive manufacturing. *Polym. Test.* 2019;73:439–46.
- [29] Lu Y, Wang Y. Monitoring temperature in additive manufacturing with physics-based compressive sensing. *J. Manuf. Syst.* 2018;48:60–70.
- [30] Lu Y, Wang Y. An efficient transient temperature monitoring of fused filament fabrication process with physics-based compressive sensing. *IIEE Trans.* 2019;51(2):168–80.
- [31] Mahmoud Y, Manoochehri S. In-situ temperature monitoring of ABS during fused filament fabrication (FFF) process with varying process parameters. In: *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers; 2021.
- [32] Williams NX, Watson N, Joh DY, Chilkoti A, Franklin AD. Aerosol jet printing of biological inks by ultrasonic delivery. *Biofabrication* 2020;12(2):025004. <https://doi.org/10.1088/1758-5090/ab5cfs>.
- [33] Jafari-Marandi R, Khanzadeh M, Tian W, Smith B, Bian L. From in-situ monitoring toward high-throughput process control: cost-driven decision-making framework for laser-based additive manufacturing. *J. Manuf. Syst.* 2019;51:29–41. <https://doi.org/10.1016/j.jmsys.2019.02.005>.
- [34] Yadav P, Rigo O, Arvieu C, Le Guen E, Lacoste E. In situ monitoring systems of the SLM process: on the need to develop machine learning models for data processing. *Crystals* 2020;10(6):524.
- [35] Jin Z, Zhang Z, Gu GX. Autonomous in-situ correction of fused deposition modeling printers using computer vision and deep learning. *Manuf. Lett.* 2019;22:11–5. <https://doi.org/10.1016/j.mfglet.2019.09.005>.
- [36] Wang Y, Huang J, Wang Y, Feng S, Peng T, Yang H, et al. A CNN-based adaptive surface monitoring system for fused deposition modeling. *IEEE/ASME Trans. Mechatron.* 2020;25(5):2287–96. <https://doi.org/10.1109/TMECH.2020.2996223>.
- [37] Rapple J, Plotkowski A, Kirka MM, Dinwiddie R, Okello A, Dehoff RR, et al. Thermographic microstructure monitoring in electron beam additive manufacturing. *Sci Rep* 2017;7(1):1–16.
- [38] Shumway RH, Stoffer DS, Stoffer DS. *Time series analysis and its applications*. Vol. 3. Springer; 2000.
- [39] Fan D, Sun H, Yao J, Zhang K, Yan X, Sun Z. Well production forecasting based on ARIMA-LSTM model considering manual operations. *Energy* 2021;220:119708. <https://doi.org/10.1016/j.energy.2020.119708>.
- [40] Choi HK. Stock price correlation coefficient prediction with ARIMA-LSTM hybrid model (arXiv preprint arXiv:1808.01560). 2018.
- [41] Liu J, Tan X, Wang Y. CSSAP: Software Aging Prediction for Cloud Services Based on ARIMA-LSTM Hybrid Model. In: 2019 IEEE International Conference on Web Services (ICWS); 2019. <https://doi.org/10.1109/ICWS.2019.00055>.
- [42] Han Y. A forecasting method of pharmaceutical sales based on ARIMA-LSTM model. In: 2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT); 2020. <https://doi.org/10.1109/ISCTT51595.2020.00064>.
- [43] Abebe M, Noh Y, Kang Y-J, Seo C, Kim D, Seo J. Ship trajectory planning for collision avoidance using hybrid ARIMA-LSTM models. *Ocean Eng* 2022;256:111527. <https://doi.org/10.1016/j.oceaneng.2022.111527>.
- [44] Wang Z, Lou Y. Hydrological time series forecast model based on wavelet denoising and ARIMA-LSTM. In: 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC); 2019. <https://doi.org/10.1109/ITNEC.2019.8729441>.
- [45] Xu D, Zhang Q, Ding Y, Zhang D. Application of a hybrid ARIMA-LSTM model based on the SPEI for drought forecasting. *Environ Sci Pollut Res* 2022;29(3):4128–44. <https://doi.org/10.1007/s11356-021-15325-z>.
- [46] Dave E, Leonardo A, Jeanice M, Hanafiah N. Forecasting Indonesia exports using a hybrid model ARIMA-LSTM. *Procedia Comput. Sci.* 2021;179:480–7. <https://doi.org/10.1016/j.procs.2021.01.031>.
- [47] Wu X, Zhou J, Yu H, Liu D, Xie K, Chen Y, et al. The development of a hybrid wavelet-ARIMA-LSTM model for precipitation amounts and drought analysis. *Atmosphere* 2021;12(1):74.
- [48] Guo S, Wang D, Chen J, Feng Z, Guo WG. Predicting nugget size of resistance spot welds using infrared thermal videos with image segmentation and convolutional neural network. *J. Manuf. Sci. Eng.* 2021;144(2). <https://doi.org/10.1115/1.4051829> (3/26/2023).
- [49] Du P, Li E, Xia J, Samat A, Bai X. Feature and model level fusion of pretrained CNN for remote sensing scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 2019;12(8):2600–11. <https://doi.org/10.1109/JSTARS.2018.2878037>.
- [50] Hoffmann H. Kernel PCA for novelty detection. *Pattern Recognition* 2007;40(3):863–74. <https://doi.org/10.1016/j.patrec.2006.07.009>.
- [51] Ahmed WS, Karim AAA. The Impact of Filter Size and Number of Filters on Classification Accuracy in CNN. In: 2020 International Conference on Computer Science and Software Engineering (CSASE); 2020. <https://doi.org/10.1109/CSASE48920.2020.9142089>.
- [52] Alpaydin E. *Introduction to Machine Learning*. MIT press; 2020.
- [53] Belaid ON, Loudini M. Classification of brain tumor by combination of pre-trained VGG16 CNN. *J. Inf. Technol. Manag.* 2020;12(2):13–25. <https://doi.org/10.22059/jitm.2020.75788>.
- [54] Wu Z, Shen C, van den Hengel A. Wider or deeper: revisiting the ResNet model for visual recognition. *Pattern Recognit.* 2019;90:119–33. <https://doi.org/10.1016/j.patrec.2019.01.006>.
- [55] Zhu Y, Newsam S. DenseNet for dense flow. In: 2017 IEEE International Conference on Image Processing (ICIP); 2017. <https://doi.org/10.1109/ICIP.2017.8296389>.
- [56] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vision* 2015;115(3):211–52. <https://doi.org/10.1007/s11263-015-0816-y>.
- [57] Nebauer C. Evaluation of convolutional neural networks for visual recognition. *IEEE Trans Neural Netw* 1998;9(4):685–96. <https://doi.org/10.1109/72.701181>.
- [58] Souleyman C, Hongxun Y, Yanfeng G, Moussa A. Deep feature extraction and combination for remote sensing image classification based on pre-trained CNN models. In: *Proc.SPIE*; 2017. <https://doi.org/10.1117/12.2281755>.
- [59] Souleyman C, Hongxun Y, Yanfeng G, and Moussa A. "Deep feature extraction and combination for remote sensing image classification based on pre-trained CNN models". In: *Proc.SPIE*. 2017. <https://doi.org/10.1117/12.2281755>.
- [60] Schölkopf B, Smola A, Müller K-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput* 1998;10(5):1299–319. <https://doi.org/10.1162/089976698300017467>.
- [61] Agarwal S, Vijaya Saradhi V, Karnick H. Kernel-based online machine learning and support vector reduction. *Neurocomputing* 2008;71(7):1230–7. <https://doi.org/10.1016/j.neucom.2007.11.023>.
- [62] Vert J-P, Tsuda K, Schölkopf B. A primer on kernel methods. *Kernel methods in computational biology* 2004;47:35–70.
- [63] Van Houdt G, Mosquera C, Nápoles G. A review on the long short-term memory model. *Artif. Intell. Rev.* 2020;53(8):5929–55. <https://doi.org/10.1007/s10462-020-09838-1>.
- [64] Zhu X, Li L, Liu J, Li Z, Peng H, Niu X. Image captioning with triple-attention and stack parallel LSTM. *Neurocomputing* 2018;319:55–65. <https://doi.org/10.1016/j.neucom.2018.08.069>.
- [65] Guo S, Wang D, Chen J, Feng Z, Guo W. Learning the Temporal Effect in Infrared Thermal Videos With Long Short-Term Memory for Quality Prediction in Resistance Spot Welding. In: *Proceedings of the ASME 2022 17th International Manufacturing Science and Engineering Conference*. Vol. 2. West Lafayette, Indiana, USA: Manufacturing Processes; Manufacturing Systems; 2022. <https://doi.org/10.1115/MSEC2022-85422>. June 27–July 1, 2022. V002T05A035. ASME.
- [66] Bottou L. Stochastic gradient learning in neural networks. *Proceedings of Neuro-Nimes* 1991;91(8):12.
- [67] Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and*

- Knowledge-Based Systems 1998;06(02):107–16. <https://doi.org/10.1142/S0218488598000094> (2021/11/01).
- [68] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput* 1997;9(8): 1735–80. <https://doi.org/10.1162/neco.1997.9.8.1735>.
- [69] Li K, Zhao X, Bian J, Tan M. Sequential learning for multimodal 3D human activity recognition with Long-Short Term Memory. In: 2017 IEEE International Conference on Mechatronics and Automation (ICMA); 2017. <https://doi.org/10.1109/ICMA.2017.8016048>.
- [70] Matrix Theory and Applications. In: Johns Charles R, editor. The hadamard product" by Roger A. Horn, Proceedings of Symposia in Applied Mathematics. Vol. 40. Phoenix, AZ: American Mathematical Society; 1989. p. 87–170. Jan. 10–11, 1989.
- [71] Suzuki K. Overview of deep learning in medical imaging. *Radiol Phys Technol* 2017;10(3):257–73.
- [72] McDonald GC. Ridge regression. In: WIREs Computational Statistics. 1(1); 2009. p. 93–100. <https://doi.org/10.1002/wics.14>. 2022/11/28.
- [73] Bozdogan H. Akaike's information criterion and recent developments in information complexity. *J. Math. Psychol.* 2000;44(1):62–91. <https://doi.org/10.1006/jmps.1999.1277>.
- [74] Hyndman RJ, Athanasopoulos G. Forecasting: principles and practice. In: OTexts; 2018.
- [75] Rodriguez JD, Perez A, Lozano JA. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE Trans Pattern Anal Mach Intell* 2010;32(3): 569–75. <https://doi.org/10.1109/TPAMI.2009.187>.
- [76] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017.
- [77] Atila O, Şengür A. Attention guided 3D CNN-LSTM model for accurate speech based emotion recognition. *Appl Acoust* 2021;182:108260. <https://doi.org/10.1016/j.apacoust.2021.108260>.
- [78] Akilan T, Wu QJ, Safaei A, Huo J, Yang Y. A 3D CNN-LSTM-based image-to-image foreground segmentation. *IEEE Transactions on Intelligent Transportation Systems* 2020;21(3):959–71. <https://doi.org/10.1109/TITS.2019.2900426>.