MACHINE LEARNING FOR MODELING OF PRINTING SPEED IN CONTINUOUS PROJECTION STEREOLITHOGRAPHY

Haiyang He[a, b], Yang Yang[c], Yayue Pan[a]*

[a]Department of Mechanical and Industrial Engineering
University of Illinois at Chicago, 2039 Engineering Research Facility
842 W. Taylor Street (M/C 251)
Chicago, Illinois 60607, United States

[b]Department of Computer Science
University of Illinois at Chicago, Room 1120 SEO
851 S. Morgan (M/C 152)
Chicago, Illinois 60607, United States

[c] PPDAI GROUP INC.
999 Dangui Road, Pudong District
Shanghai 201203, China

Abstract

Continuous projection stereolithography technologies, also known as the Continuous Liquid Interface Production (CLIP), can achieve build speeds an order of magnitude faster than conventional layer-by-layer stereolithography process. However, identification of the proper continuous printing speed remains a grand challenge in the process planning. To successfully print a part continuously, the printing speed needs to be carefully adjusted and calibrated for the given geometry. In this paper, we investigate machine learning techniques for modeling and predicting the proper printing speed in the continuous projection stereolithography process. The synthetic dataset is generated by physics-based simulations. Experimental dataset is constructed for training the machine learning models to find the appropriate speed range and the optimum speed. Conventional machine learning techniques including Decision Tree, Naïve Bayes, Nearest Neighbors, and Support Vector Machine (SVM), ensemble methods including Random Forest, Gradient Boosting, and Adaboosting, and the deep learning approach Siamese Network are tested and compared. Experimental results validate the effectiveness of these machine learning models and show that the Siamese Network model gives the highest accuracy.

KEYWORDS
Continuous Projection Stereolithography; CLIP; Machine Learning; Deep Neural Network; Siamese Network; Continuous Printing Speed

Introduction

Continuous projection stereolithography is a vat photopolymerization based additive manufacturing technology that does not involve discrete layers because the curing part is drawn out of the resin without interruption [1, 2], as illustrated in Fig. 1. It is also known as Continuous Liquid Interface Production (CLIP) and orders of magnitude faster than the layer-by-layer photopolymerization have been demonstrated in Tumbleston *et al.* work [1] and He *et al* work [2]. A key factor to the success of continuous printing is a proper continuous elevation speed $\boldsymbol{V}$ as shown in Fig.1, which is, however, challenging to identify. An over fast speed will result in the failure of bonding newly cured material to the part. On the other hand, an over slow speed tends to lead to the adhesion between the part and the oxygen permeable window. Current approaches to searching for the proper continuous elevation speed mainly rely on empirical knowledge gained from trial and error experiments. Since the working continuous elevation speed varies with the printing geometry, there is an urgent need to develop a systematic and fundamental approach to replace the current trail-error method for identifying the proper speed.

To address this challenge, this work focuses on investigating machine learning techniques for the continuous printing speed modeling, selection, and optimization.

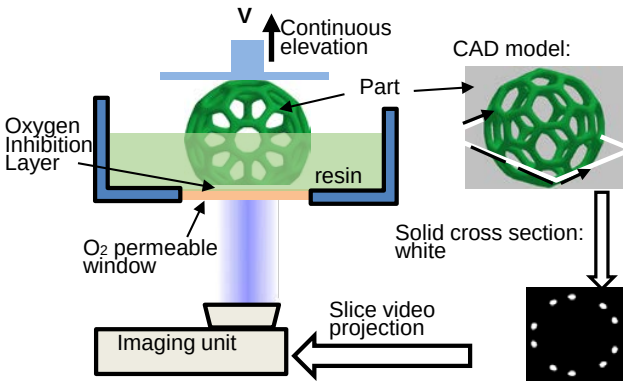


Fig. 1 Illustration of continuous printing with oxygen permeable window

Machine learning has been proved effective in various manufacturing systems. In [3], advances and trends in cyber-physical manufacturing systems have been discussed. It reviews applications of big data analytics in manufacturing systems. An overview of additive manufacturing informatics has been provided in [4], which emphasizes the importance of integrating data mining and additive manufacturing systems. The advantages, applications, and technology progress of AM, AM data and big data analytics for AM are presented in [5]. The study of data mining in selective laser melting (SLM) sensor data is conducted in [6]. Data-driven surrogate models are proposed in [7] to identify important variables and find appropriate process parameters in SLM. In [8], a data-enabled interferometric curing monitoring and measuring model is proposed to estimate the height profile of cured parts and realize the desired real-time measurement for polymer additive manufacturing processes. Despite those advances of machine learning in manufacturing, its application in continuous projection stereolithography has not been investigated yet. In this paper, we aim to investigate various machine learning techniques for printing speed selection and optimization in continuous projection stereolithography.

Due to the large design space of parameters for continuous printing, it is critical to first identify a reasonable speed range and then identify the optimum speed. Previous studies [9-11] have shown that the separation force incurred in the process is closely related to the manufacturing speed. In addition, the separation force is a strong indicator and a readily measurable signal of whether the printing is successful. Therefore, the magnitude of separation force in a certain printing process can be used to evaluate the current condition or predict the outcome of the process. To take advantage of this relationship between separation force and the printing process, a theoretical model of separation force has been utilized as a heuristic for determining the initial proper speed range. A synthetic dataset is generated by enumerating the exhaustive combination of various levels for each factor in the theoretical model. A large synthetic dataset is obtained by considering various levels of different parameters. Prior information is extracted from it when no experimental data is available. Thus, when initiating tests for a new experimental design with little prior knowledge, machine learning models could be first trained on this synthetic dataset to predict the outcome of experimental designs. Experiments can be carried out selectively and effectively based on the predicted outcome. In addition, experimental data would be collected and added to a dynamically growing experimental dataset. To improve accuracy, machine learning models will be trained on newly updated experimental dataset and improved.

2 Theoretical Modelling, Data Collection and Model Introduction

2.1 Theoretical Modelling of the Separation Process

The separation of the newly cured layer from the constrained surface together with the liquid filling induced by this separation process is a significant procedure of the constrained surface projection SL technique.

The force incurred in this separation process is a strong indicator for the success of the printing, as illustrated in equation (1), where $F_{separation}$ denotes the maximum separation force for printing a certain layer, F_0 and F_1 are the lower bound and upper bound of the reasonable separation force range for printing that layer, respectively. An over large separation force usually implies the adhesion between the printed part and the constrained surface. While an over small separation force commonly relates to the failure of bonding newly cured layer to the printed part.

$$\begin{cases} F_{separation} \subset [F_0, F_1], & success \\ F_{separation} \not\subset [F_0, F_1], & fail \end{cases} \tag{1}$$

Our previous study has modelled the separation force for smooth constrained surface and pressure drop for textured constrained surface by concerning the separation mechanism and liquid filling effects around the separation interface, as shown in the following equations [9, 12, 13]:

$$F_1 = \frac{3\pi \cdot \mu V}{2 \cdot h^3} \cdot R^4 \tag{2}$$

$$\frac{dp}{dr} = V \cdot \pi r^2 \cdot 2\mu \cdot \frac{\left(4\pi \cdot r + \frac{2nd}{cos\alpha} - n \cdot w\right)^2}{(2\pi \cdot r \cdot h + 0.5n \cdot w \cdot d)^3} \tag{3}$$

where V is the separation speed, r is a variable ranging from 0 to R, which is the radius of the part cross section, n denotes the number of grooves of the micro texture, μ represents the viscosity of the resin, w and d are the width and depth of the grooves of the micro texture. h denotes the height of the initial gap, which is the oxygen inhibition layer thickness.

Two constrained surfaces, island surface (IS) and textured surface(TS) are investigated for continuous printing. The surface of IS is smooth, so equation (2) is used for calculating the separation force. TS implements a microtextured surface and equation (3) is applied. By adding terms compensating factors such as the plasticity and deformation of constrained surface, platform initial position calibration, and random noises to above equations, the calibrated equation for separation force for IC and TC are modified as follows:

$$\begin{cases} F_{IC} = k_1 \cdot \frac{3\pi \cdot \mu V}{2 \cdot h^3} \cdot R^4 + k_2 \\ F_{TC} = k_3 \cdot \int_0^R \frac{dp}{dr} dr + k_4 \end{cases} \tag{4}$$

where k_1 is calibrated to be 0.027 and k_3 0.03, mainly for compensating the PDMS deformation, k_2 and k_4 are randomly generated noises with an absolute value of less than 0.02N according to empirical experience.

2.2 Synthetic and Experimental Dataset

2.2.1 Synthetic data generation

Based on the theoretical modelling of the separation force, simulation has been performed by considering nine process parameters, including resin viscosity, cross section size of the part, manufacturing velocity, PDMS thickness, constrained surface type, duration of frame, video projection time, micro texture groove width, micro texture groove depth etc. Each of these process parameters has several levels, which represent some representative parameter values in the real manufacturing processes. An example of the parameters with their corresponding values are given in Table 1.

Table 1 Various levels of different process parameters

Process parameters	Levels		
Resin viscosity ($Pa \cdot s$)	0.09	0.12	0.14
Geometry size (mm^2)	3.14	12.56	50.24

Manufacturing velocity (mm/s)	0.025			0.038			0.05		
PDMS thickness (mm)	1			2			4		
Constrained surface type	Smooth			Textured			Island		
Duration of frame(s)	0.5			1			1.5		
Video projection time (min)	15			20			30		
Groove width (μm)	100								
Groove depth (μm)	100								
Cross section size j (mm^2)	3.1	7.1	12.6	19.6	28.3	38.5	50.3	63.6	78.5

Combinations of different levels for these parameters will result in different sets of experiments. The corresponding separation force is calculated using Equation (4). In total, 6000 instances are generated in the simulated dataset. Threshold F_{0_j} and F_{1_j}, which are the proper lower bound and upper bound of empirical separation force, are set to generate the corresponding labels for each piece of data. In F_{0_j} and F_{1_j}, j is the index for distinguishing parts of different sizes. i is the instance index and $y_{i,i}$ is used as the label for each piece of data, with a value of 1 means the part with a size in j level is successfully printed and 0 denotes a failure:

$$y_{j,i} = \begin{cases} 1, & if\ \ F_{0_j} < \mathrm{F}_{\text{separation}} < F_{1_j} \\ 0, & if\ \ \mathrm{F}_{\text{separation}} > F_{0_j}\ or\ \ \mathrm{F}_{\text{separation}} < F_{1_j} \end{cases}$$

If the calculated separation force is larger than the upper bound threshold F_{0_j} or smaller than the lower bound threshold F_{1_j}, then the printing part is failed, and a 0 label will be assigned to the data. The number of instances of 0 label and 1 label are close in the simulated dataset. Therefore, the dataset is balanced, and no oversampling and under-sampling is needed. Machine learning techniques are trained on this simulation generated dataset and the learned models provide some prior knowledge before designed experiments are carried out. To test the effectiveness of the machine learning models trained on this synthetic dataset, the learned models are evaluated on experimental obtained data.

2.2.2 Experimental data collection

Experiments are carried out to test different speeds for continuous printing. The schematic of the setup is given in Fig. 2. The image of the continuous projection SL system is shown in Fig. 3. A precision position stage from Velmex is used as the Z stage. A process control testbed has been developed using C++ language. It integrates the geometry slicing, image projection, and motion controlling. A KMotion control board from DYNOMOTION is implemented to control the Z stage motion, light projection and synchronized the motion and projection. An online force monitoring unit has been developed in Matlab/Simulink for measuring the separation force. A load cell (LRM 200 from Futek) is utilized to collect the force signals, together with a data acquisition (DAQ) device (USB 6008 from National Instruments). The online force monitoring system reads and processes data from the load cell, and records separation force measured in real time during the continuous manufacturing processes. Two commercial photopolymers, PerfactoryTM LS600M from EnvisionTEC Inc. and G+ (green) from MakeJuice Labs are used as the materials in the experiments.

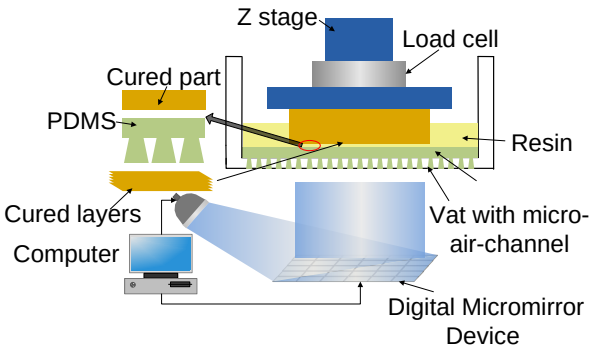


Fig. 2 Schematic of the experiment setup

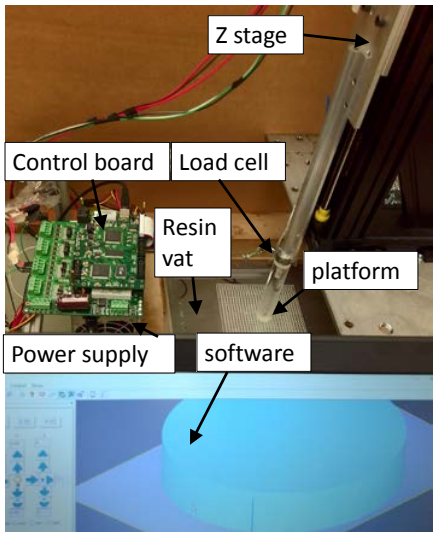


Fig. 3 Image of experimental setup

Parameters described in the mathematical model, which are readily accessible through measurements, are selected as the variables in the experiments and the other conditions are fixed. For simplicity, cylinders of different diameters and heights are printed. Each set of process parameters and the corresponding outcomes are recorded and summarized to generate an experimental dataset. The label is 1 if the part is successfully printed (without consideration of the surface roughness) and 0 otherwise. An initial experimental dataset of 180 instances is collected. Machine learning algorithms are trained and tested on this experimentally obtained dataset to find the proper manufacturing speed for different sets of process parameters. The collected experimental dataset is balanced with the same number of successful and failure prints (positive and negative labels). Fig. 4 gives an example of samples continuously printed with various speed.

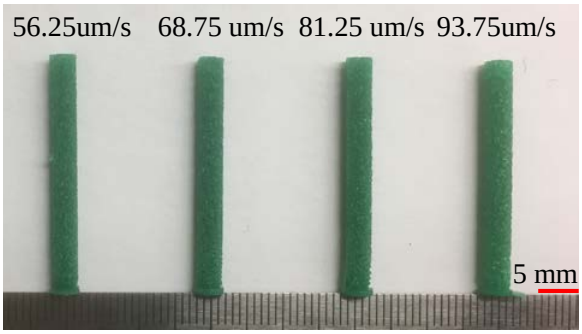


Fig. 4 Cylinder samples continuously printed under different speed

2.3 Machine Learning Models

2.3.1 Conventional techniques

To classify the simulated data, conventional approaches, including K Nearest Neighbours (KNN), Support Vector Machine (SVM), Decision Tree, Logistic Regression, Quadratic Discriminant Analysis (QDA), Gaussian Processes (GP), NaiveBayes and Neural Network are implemented first with scikit-learn using Python. KNN implements 5-nearest neighbours vote and uniform weights are used for all points in each neighbourhood. The implementation of SVM is based on libsvm, commonly used radial basis function is selected as kernel function and all classes are supposed to have weight one. The split criterion for decision tree is Gini impurity and all classes are equally weighted to be one. The regularization utilized in logistic regression is l2 penalty, which adds "squared magnitude" of coefficient as penalty term to the loss function, and the tolerance for stopping criteria is 1e-4. 'Liblinear' algorithm is applied in the optimization problem. The maximum number of iterations taken for the solvers to converge is 100. Quadratic Discriminant Analysis (QDA) is a classifier with a quadratic decision boundary, generated by fitting class conditional densities to the data and using Bayes' rule. Gaussian process classification (GPC) is based on Laplace approximation, the 'fmin_l_bfgs_b' algorithm from scipy.optimize is used as the optimizer. Naïve Bayes is implemented using the Gaussian Naive Bayes algorithm and the likelihood of the features is assumed to be Gaussian. The neural network model is implemented using multi-layer perceptron classifier. Rectified linear unit function (ReLU) is used as the activation function, and 'adam' is selected as the optimizer.

2.3.2 Ensemble Methods

The goal of ensemble methods is to combine the predictions of several base estimators built with a given learning algorithm to improve the generalizability and robustness over a single estimator. Two families of ensemble methods are usually distinguished: One is the averaging methods, which is to build several estimators independently and then to average their predictions. This section implements one of the most representative averaging approaches, Random Forests. The other is boosting methods, which is to combine several weak models to produce a powerful ensemble. Ada Boost and Gradient Tree Boosting are utilized as representatives of boosting methods in this section [14]. Random Forest is implemented with 25 trees. The split criterion for each tree is Gini impurity and all classes are equally weighted to be one. The base estimator for Ada Boost is decision tree. The maximum number of estimators is 50 and the learning rate is 1. For gradient boosting, 'deviance' is used as the loss function, the learning rate is 0.1 and the number of boosting stages to perform is 100.

2.3.3 Siamese Network

Deep neural networks are currently very popular in machine learning community because they can have arbitrarily large number of trainable parameters and usually achieve satisfactory results [15]. However, it requires a large amount of data to train the parameters, which is sometimes not available for manufacturing systems. This becomes more serious when it comes to collecting data obtained from relatively time consuming and expensive experiments. "Siamese" neural network, which was introduced by LeCun [16], only requires just one training example of each interested class. This network can still be trained with many data points, as long as they are in the similar domain to other training data points. Siamese Network was first developed for signature verification written on pen-input tablet. It consists of two identical sub-networks with shared weights jointed at their outputs, as illustrated in Fig. 5. Given a pair of signature images, these two sub-networks extract features from one image from the pair and pass the learned information through a contrastive loss function to measure their distance. The contrastive loss function can be described as follows [17]:

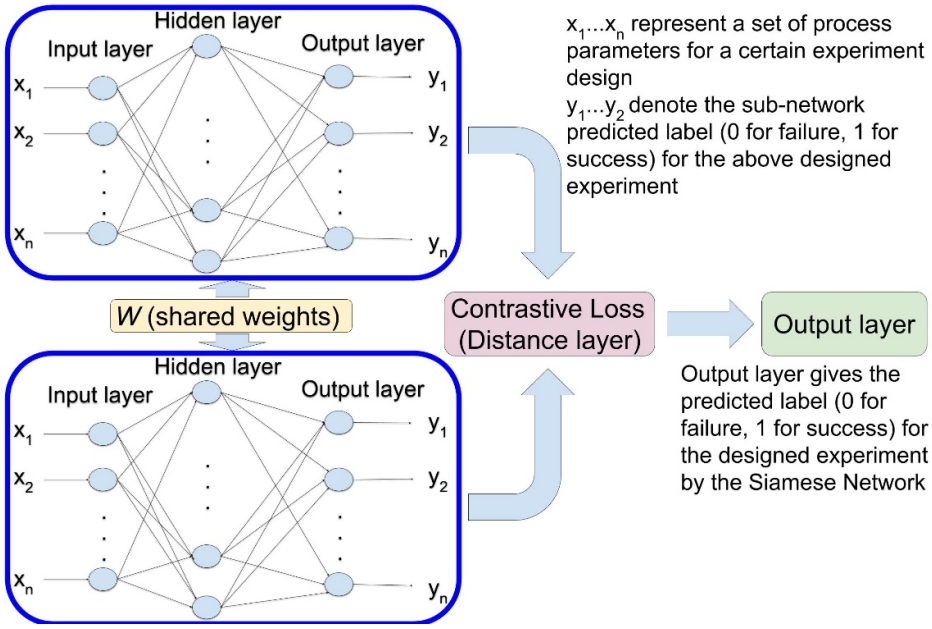
$$(1-Y)\frac{1}{2}(D_w)^2 + (Y)\frac{1}{2}\{max(0, m - D_w)\}^2 \quad (1)$$

Y=0 if the pair of images are similar, and Y=1 if they are dissimilar. The parameterized Euclidean distance function between these two images D_w is defined as:

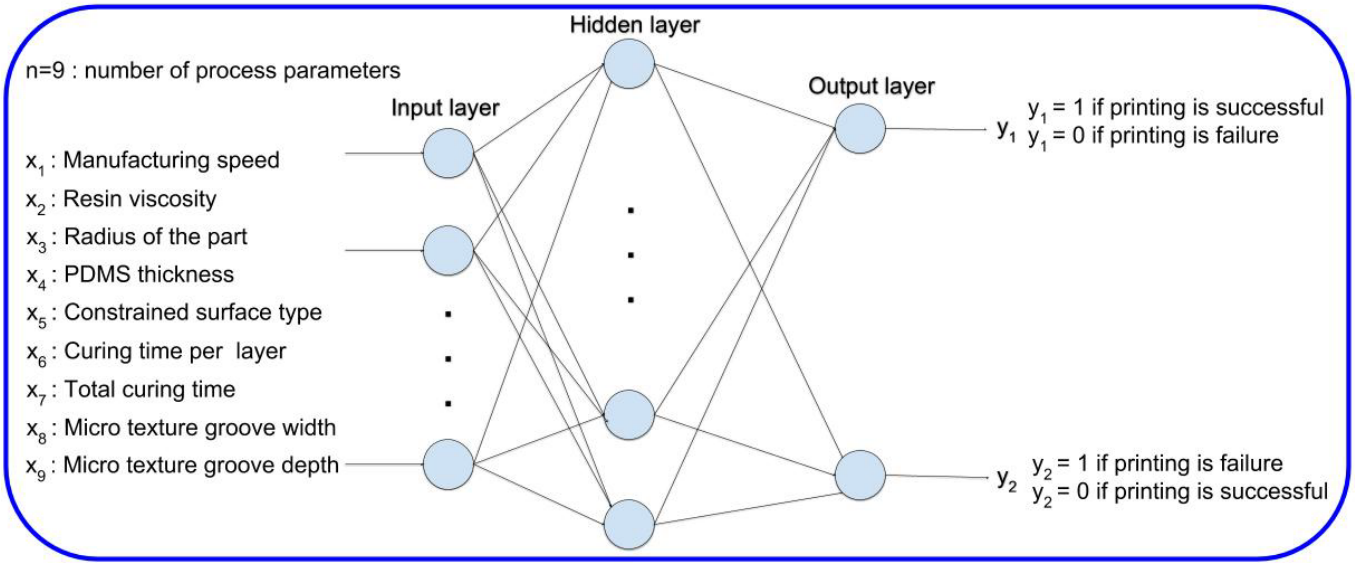
$$D_w(X_1, X_2) = \sqrt{G_W(X_1) - G_W(X_2)} \quad (2)$$

where X_1, X_2 are a pair of image instances and G_W is the output of each individual neural network.

Intuitively, Siamese Network trained two identical neural networks using pairs of signature images, extracted feature vectors and stored the learned weights. At test time, the distance between a test signature and a known signature is calculated. Similar signatures are accepted, and forgeries are rejected.



(a) General Siamese Network



(b) Subnetwork of a Siamese Network for continuous printing

Fig. 5 Diagram of Siamese Network

The sub-network for Siamese Network in above signature verification situation is a time delay neural network. In [17], convolutional neural networks are used as sub-networks of Siamese Network for dimensionality reduction by learning an invariant mapping. A variation of deep Siamese Networks is proposed in [18]. It consists of a sequence of convolutional layers, applies ReLU activation functions and imposes a regularized cross-entropy objective to the binary classifier for one-shot image recognition. In [19], 3-layer MLP (Multilayer Perceptron) is used as the sub-network. Tanh is the activation function and the loss function is modified to be Triangular Similarity Metric Learning (TSML) objective function for dimensionality reduction and face identification. To our best knowledge, all existing applications of Siamese Networks are related to image classifications. Since Siamese Network works well on small dataset which matches the typical size of our experimental collected

dataset, a customized Siamese Network is created according to the characteristics of our dataset. To implement our own Siamese Network, base networks of 4 MLP with dropout rate 0.1 are created and ReLU is selected as the activation function after comparing it with tanh and sigmoid. Various sets of process parameters are fed into the input layer and their corresponding outcomes are collected in the output layer in the form of predicted labels. There are 32 neurons in all hidden layers except the first and last ones, which have 8 neurons. After outperforming Adagrad and RMSprop, Stochastic gradient descent with a learning rate of 0.00035 is used as the optimizer. The network is initialized with 0 mean and 0.05 standard deviation.

3 Results and discussions

3.1 Evaluation of Models Trained Using Synthetic Data

Theoretical modelling generated synthetic data can be useful when initiating a new set of experiments without any prior experimental data. The trained machine learning models on synthetic data can be used to give relatively simple while constructive predictions on designed experiments. To evaluate the performance of these models, experimental data is used for testing.

3.1.1 Evaluation of Conventional Techniques

Conventional machine learning models described in the section above are trained using the synthetic dataset. All the parameters such as learning rate, momentum, and batch size, etc., adopt the classifiers' default value. The results are shown in Table 2. Although the accuracy of these models on training set is high, their performance on testing dataset is relatively low. This cannot be explained by overfitting because no overfitting problem has been identified when training the models. The following causes may lead to the low testing accuracy: 1. The limitation of only using separation force as the evaluation metric of whether printing is successful or not; 2. The deficiency of extending the generalized separation force model developed for layer-by-layer printing to continuous printing; 3. The simulation data and experimental data are drawn from different distributions.

Table 2 Results of conventional techniques trained on synthetic dataset

Models	Training accuracy	Testing accuracy
Nearest Neighbours	0.96	0.53
QDA	0.88	0.53
Gaussian Process	0.95	0.53
Logistic Regression	0.78	0.53
Naive Bayes	0.75	0.53
Decision Tree	1	0.52
SVM	0.94	0.51
Neural Net	0.93	0.50

3.1.2 Evaluation of Ensemble Methods

Similar problems happen to the ensemble methods. The training accuracy on synthetic data and testing accuracy on experimental data are listed in Table 3. The Ada boost method outperforms other ensemble methods on the experimental dataset with an accuracy of 55.8%.

Table 3 Results of ensemble methods trained on synthetic dataset

Models	Training accuracy	Testing accuracy
Random Forest	1	0.52
Ada Boost	0.76	0.56
Gradient Boosting	0.99	0.54

3.1.3 Evaluation of Siamese Network

Although Siamese Network also suffers from the aforementioned problems, the average training accuracy of our Siamese Network on the synthetic dataset is 84.24 and the average testing accuracy on experimental dataset is 64.29, which outperforms all conventional approaches and ensemble methods mentioned above. Therefore, Siamese Network is relatively reliable for providing basic experimental guidance when only synthetic dataset is available, thus it is more suitable for designing and initializing experiments.

3.1.4 Overall Comparison of Models Trained Using Synthetic Dataset

The performance of best conventional Nearest Neighbors model and Ada Boost model, which represent the highest accuracy of conventional models and ensemble models trained on synthetic dataset, is compared with that of Siamese Network. Results are plotted in Fig. 6. It is apparent that Siamese Network can extract more information from and make better use of the synthetic dataset. This is mainly because of the contrastive loss used in Siamese Network, which can better capture the discrepancy between data from distinct classes. Therefore, Siamese Network can provide more prior knowledge when only the synthetic dataset is available.

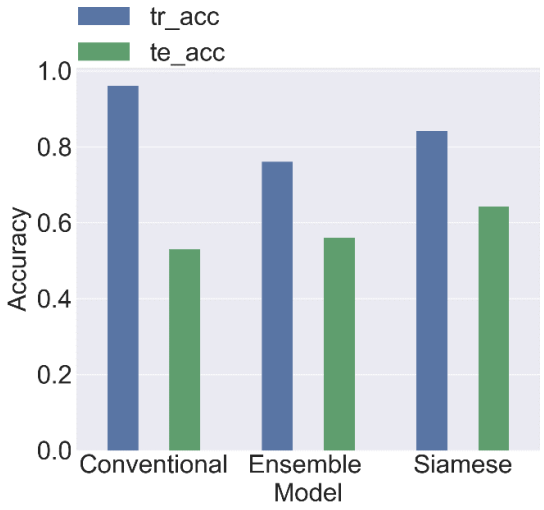


Fig. 6 Comparison of conventional methods, ensemble approaches and Siamese Network trained using synthetic dataset

3.2 Evaluation of Models Trained Using Experimental Data

3.2.1 Evaluation of Conventional Techniques

Similar to the cases using synthetic dataset, conventional approaches including K Nearest Neighbours (KNN), Support Vector Machine (SVM), Decision Tree, Logistic Regression, Quadratic Discriminant Analysis (QDA), Gaussian Processes (GP), Naïve Bayes, and Neural Network are implemented with scikit-learn using Python. All the parameter settings are set to obtain the models' best performance. To avoid overfitting, 10-fold stratified cross validation is applied to all the classifiers. The average accuracy of conventional models on training and testing data is shown in Table 4. It can be seen that Nearest Neighbors works the best among all the conventional techniques with a testing accuracy of 85.87% and Gaussian Process also works well with a testing accuracy of 85.29%.

Table 4 Results of conventional methods on experimental dataset

Models	Training accuracy	Testing accuracy
Nearest Neighbours	0.88	0.86
QDA	0.76	0.75

Gaussian Process	0.92	0.85
Logistic Regression	0.72	0.72
Naïve Bayes	0.68	0.61
Decision Tree	0.99	0.81
SVM	0.86	0.83
Neural Network	0.91	0.82

The overall performance of all conventional techniques on the testing dataset is shown in Fig. 7. The error bars depict the mean and variation of training and testing accuracy for each model on all the 10 folds of data. The blue bars show the training accuracy and the green bar denotes the testing accuracy. On the x axis, Neural N., Gaussian P., QDA, Logistic R., D. Tree, SVM, N. Bayes and N. N. represent Neural Network, Gaussian Process, Quadratic Discriminant Analysis, Logistic Regression, Decision Tree, Support Vector Machine, Naïve Bayes, and Nearest Neighbours, respectively. The same abbreviations are used in the following sections.

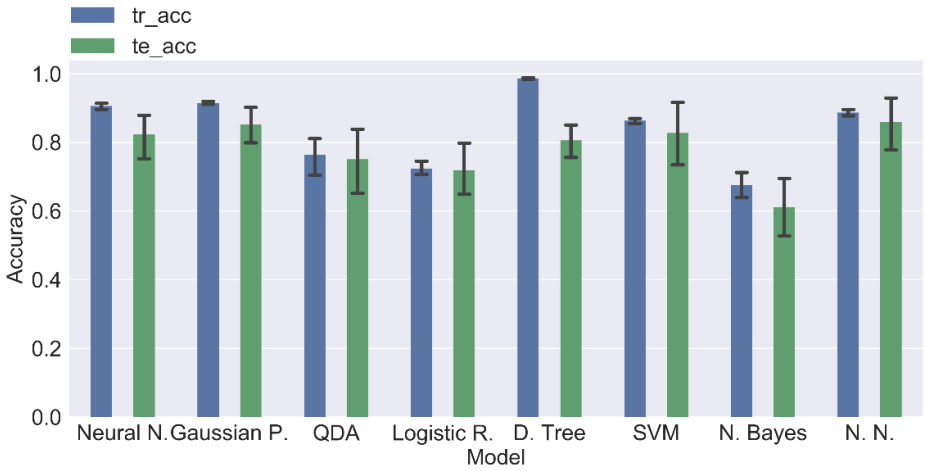


Fig. 7 Overall performance of all conventional techniques

The training and testing accuracy of Nearest Neighbors model, which works best among all conventional approaches, on each of the 10 folds of data is illustrated using blue bars and green bars respectively in Fig. 8.

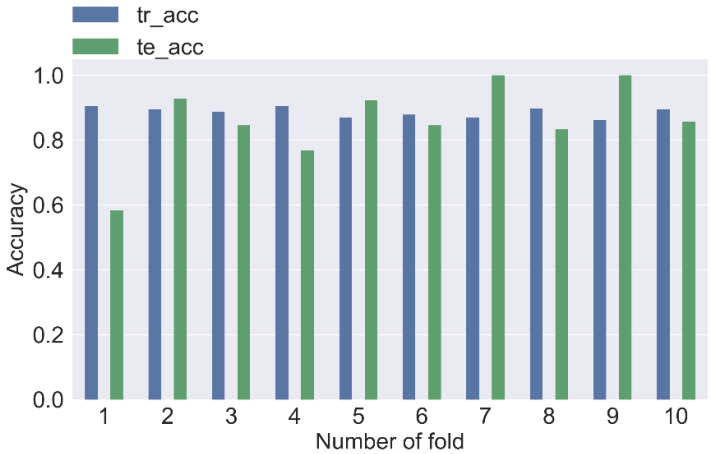


Fig. 8 Training and testing accuracy of Nearest Neighbors model on all the 10 folds of data

3.2.2 Evaluation of Ensemble Methods

Ensemble methods including Random Forests, Ada Boost, and Gradient Tree Boosting are tested on the experimental dataset. These methods are adjusted to achieve their best performance. 10-fold stratified cross validation is applied to all the classifiers. The average accuracy of conventional models on training and testing

data is shown in Table 5. Random Forest works the best among all the ensemble methods with a testing accuracy of 83.02%.

Table 5 Results of ensemble methods trained on synthetic dataset

Models	Training accuracy	Testing accuracy
Random Forest	0.88	0.83
Ada Boost	0.80	0.82
Gradient Boosting	0.88	0.80

The overall performance of all ensemble methods on each fold of the dataset is shown in Fig. 9. The error bars depict the mean and variation of training and testing accuracy for each model on all the 10 folds of data. The blue bars show the training accuracy and the green bars denote the testing accuracy. On the x axis, G. Boosting, R. Forest, and AdaB. denote Gradient Boosting, Random Forest, and Ada Boost, respectively. The same abbreviations are implemented in the following sections.

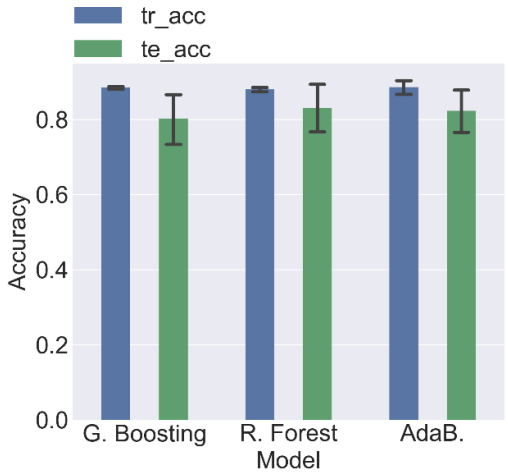


Fig. 9 Overall performance of ensemble methods

The training and testing accuracy of Random Forest model, which performs the best among the ensemble models, on each of the 10 folds of data is illustrated in Fig. 10 using blue bars and green bars respectively.

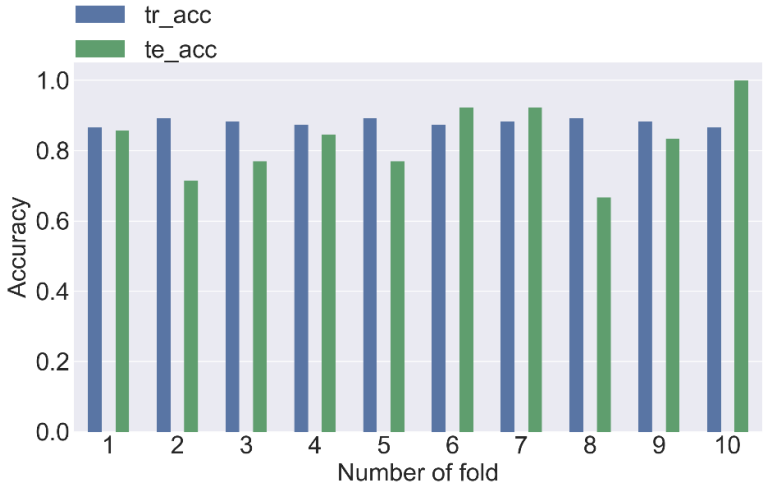


Fig. 10 Training and testing accuracy of Random Forest on all the 10 folds of data

3.2.3 Evaluation of Siamese Network

Siamese Network is also evaluated on the same experimental dataset. 10-fold stratified cross validation is also implemented. The training and testing accuracy of Siamese Network on each of the 10 folds of data is plotted in Fig. 11 using blue bars and green bars, respectively. The average training accuracy is 90.17% and the testing accuracy is 88.42%.

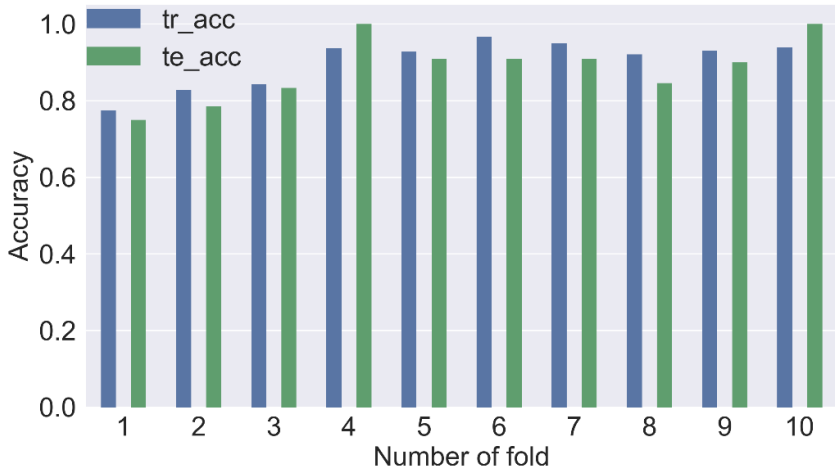


Fig. 11 Training and testing accuracy of Siamese Network on all the 10 folds of data

3.2.4 Overall Comparison of Models Trained Using Experimental Dataset

To make a better comparison, Nearest Neighbors model, which works best among all conventional approaches, is selected as the representative of conventional models. Random Forest is picked on behalf of ensemble methods. Their training and testing accuracy are compared with Siamese Network and the comparison is shown in Fig. 12. Siamese Network slightly outperforms all the other mentioned approaches. It is verified that the trained model can be used for predicting the outcome of future trials to guide the design of experiments.

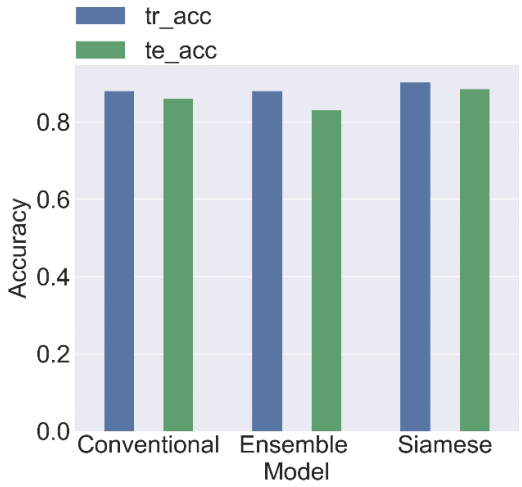


Fig. 12 Overall comparison of the models on experimental data

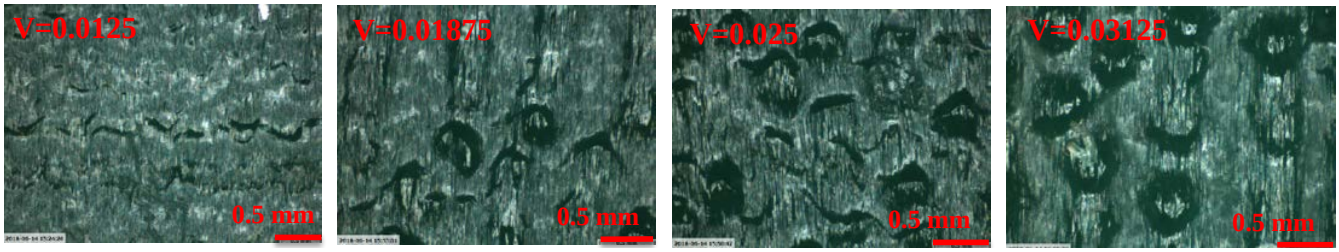
3.3 Printing Speed Optimization

The learned machine learning models in Section 3.2 can more accurately predict whether the printing speed is appropriate to print a part continuously. However, the surface quality of the printed parts varies with the continuous printing speed. the printed surface is much coarser if the filling resin is rapid when a higher continuous elevation speed is used. A tradeoff between the printing speed and the printed surface quality needs to be made.

This section investigates the application of machine learning models on identifying the optimum continuous printing speed that can produce an acceptable surface quality.

3.3.1 Data collection

To collect data, squares with the same area while different aspect ratios are continuously printed using different continuous elevation speed. The data is labeled 0 if the printing fails. The successfully printed parts are observed under microscope to check the surface quality. Since there are only a few optimum speeds for each geometry, to avoid unbalanced data, the printing with an optimum speed is labeled 3 and the rest unlabeled data are rated 1 or 2 by concerning both the surface quality and manufacturing speed. Microscopic images of some surface of printed samples are shown in Fig. 13. The average size of the pores on the printing surface is used as the surface quality evaluation metric. In this example, Fig. 13 (b) is graded as 3, (a) and (c) are rated to be 2 and (d) is marked as 1. Therefore, there are four classes in the dataset and the task becomes more challenging than previous. Due to the high cost and time-consuming characteristics of continuous printing experiments, limited data (~80) is collected.



(a) V = 0.0125 mm/s (b) V = 0.01875 mm/s (c) V = 0.025 mm/s (d) V = 0.03125mm/s

Fig. 13 Microscopic images of surface of printed 2mm x 8mm square

3.3.2 Evaluation of Conventional Techniques on Optimizing Printing Speed

Conventional approaches including K Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree, Logistic Regression, Gaussian Processes (GP) and Neural Network are tested on the dataset. To avoid overfitting, 10-fold stratified cross validation is utilized. The average accuracy of conventional models on training and testing data is shown in Table 6. Nearest Neighbours works the best among all the conventional techniques with a testing accuracy of 59%.

Table 6 Results of conventional methods on optimizing printing speed

Models	Training accuracy	Testing accuracy
Nearest Neighbours	0.64	0.59
Gaussian Process	0.60	0.53
Logistic Regression	0.51	0.46
Naive Bayes	0.61	0.53
Decision Tree	0.53	0.46
SVM	0.51	0.50
Neural Net	0.62	0.49

The overall performance of all conventional techniques on each fold of the dataset is shown in Fig. 14. The error bars depict the mean and variation of training and testing accuracy for each model on all the 10 folds of data. The blue bars show the training accuracy and the green bars the testing accuracy.

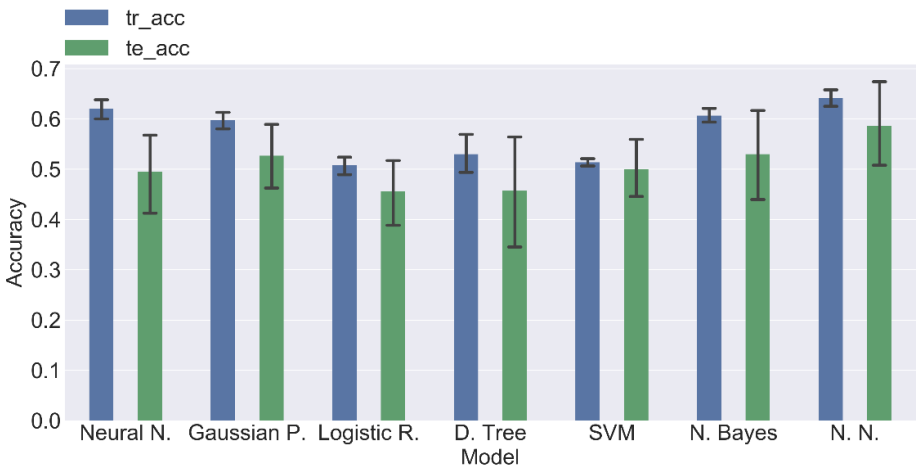


Fig. 14 Overall performance of all conventional techniques

The training and testing accuracy of Nearest Neighbours model on each of the 10 folds of data is plotted in Fig. 15 using blue bars and green bars, respectively.

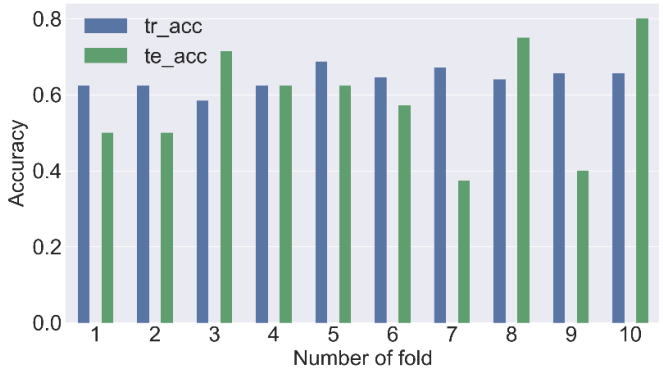


Fig. 15 Training and testing accuracy of Nearest Neighbors model on all the 10 folds of data

3.3.3 Evaluation of Ensemble methods on Optimizing Printing Speed

Ensemble methods are also tested for optimizing the printing speed. The settings of these methods are modified to achieve the best accuracy. 10-fold stratified cross validation is applied to all the classifiers. The average accuracy of ensemble methods on training and testing data is shown in Table 7. Random Forest works the best among all the ensemble methods with a testing accuracy of 73%.

Table 7 Results of ensemble methods trained on synthetic dataset

Models	Training accuracy	Testing accuracy
Random Forest	0.75	0.73
Ada Boost	0.68	0.60
Gradient Boosting	0.70	0.68

The overall performance of all ensemble methods on all 10 folds of the dataset is shown in Fig. 16. The error bars depict the mean and variation of training and testing accuracy for each model on all the 10 folds of data. The blue bars show the training accuracy and the green bars denote the testing accuracy.

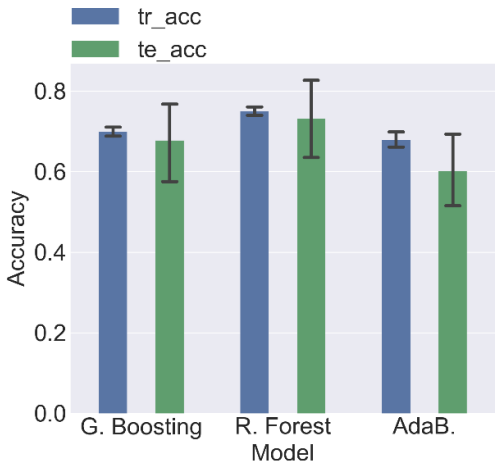


Fig. 16 Overall performance of ensemble methods

The training and testing accuracy of Random Forest model, which performs the best among the ensemble models, on each of the 10 folds of data is plotted in Fig. 17 using blue bars and green bars, respectively.

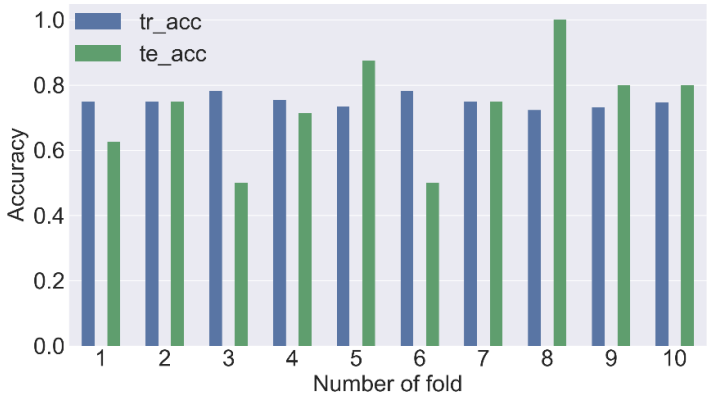


Fig. 17 Training and testing accuracy of Random Forest model on all the 10 folds of the dataset

3.3.4 Evaluation of Siamese Networks on Optimizing Printing Speed

Siamese Network is evaluated on the same experimental dataset, with a 10-fold stratified cross validation. The related training and testing accuracy on each of the 10 folds of data is plotted in Fig. 18 using blue bars and green bars, respectively. The average training accuracy is 80.14% and the testing accuracy is 82.67%, which outperforms all above mentioned approaches. Thus, it is also verified that Siamese Networks is effective on optimizing printing speed.

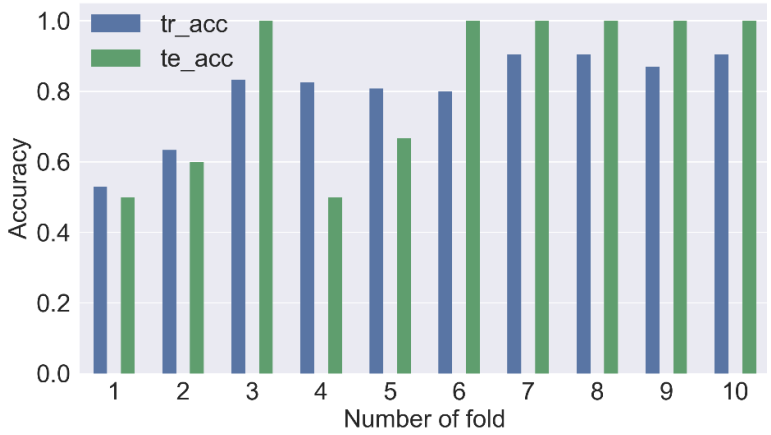


Fig. 18 Training and testing accuracy of Siamese Network on all the 10 folds of the dataset

3.3.5 Overall Comparison of Models for Speed Optimization

A comparison of conventional models (Decision Tree is used as a representative), ensemble methods (Gradient Boosting is selected as a representative), and Siamese Network for printing speed optimization is shown in Fig. 19. Siamese Network exceeds all the other approaches. The overall relatively low accuracy may due to the small size of the dataset. Nonetheless, the trained Siamese Network performs the best and shows a great potential for predicting the outcome of future trials to guide the design of printing jobs.

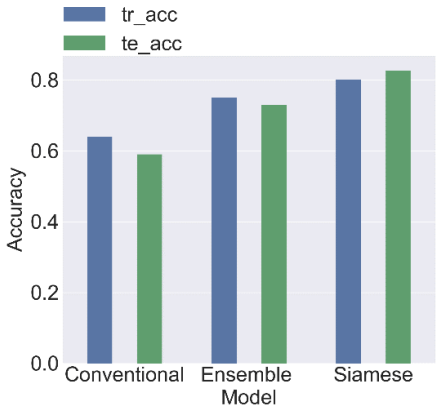


Fig. 19 Overall comparison of the models for printing speed optimization

4 Conclusions

This paper investigates the application of machine learning models in predicting feasible printing speed for Continuous Projection Stereolithography process. Conventional techniques, ensemble approaches, and Siamese Networks are investigated and compared. Siamese Network works the best among all the investigated models. It can effectively extract useful information from mathematical model generated synthetic dataset. Given an experimental dataset, Siamese Network can more accurately classify the data and give relatively more reliable guidance for future experimental design, compared to all other models. Experimental results also validated that Siamese Network is more effective on capturing features for identifying the optimum manufacturing speed. With the help of Siamese Network, a dynamically growing dataset for continuous printing can be enriched effectively, and the continuous printing process can be planned efficiently.

Acknowledgement

This work is partially supported by the National Science Foundation under Grant 1563477. The authors would like to acknowledge the funding of this project.

References

[1] Tumbleston, J.R., Shirvanyants, D., Ermoshkin, N., Janusziewicz, R., Johnson, A.R., Kelly, D., Chen, K., Pinschmidt, R., Rolland, J.P., Ermoshkin, A. and Samulski, E.T., 2015. Continuous liquid interface production of 3D objects. Science, p.aaa2397.

[2] Chen, Y., Mao, H. and Li, X., University of Southern California (USC), 2016. Mask Video Projection Based Stereolithography with Continuous Resin Flow. U.S. Patent Application 15/187,713.

[3] Lee, J., Lapira, E., Bagheri, B. and Kao, H.A., 2013. Recent advances and trends in predictive manufacturing systems in big data environment. Manufacturing Letters, 1(1), pp.38-41.

[4] Mies, D., Marsden, W. and Warde, S., 2016. Overview of additive manufacturing informatics:"a digital thread". Integrating Materials and Manufacturing Innovation, 5(1), p.6.

[5] Wang, L. and Alexander, C.A., 2016. Additive manufacturing and big data. International Journal of Mathematical, Engineering and Management Sciences, 1(3), pp.107-121.

[6] Uhlmann, E., Pontes, R.P., Laghmouchi, A. and Bergmann, A., 2017. Intelligent pattern recognition of a SLM machine process and sensor data. Procedia CIRP, 62, pp.464-469.

[7] Kamath, C., 2016. Data mining and statistical inference in selective laser melting. The International Journal of Advanced Manufacturing Technology, 86(5-8), pp.1659-1677.

[8] Zhao, X. and Rosen, D.W., 2017. A data mining approach in real-time measurement for polymer additive manufacturing process with exposure controlled projection lithography. Journal of Manufacturing Systems, 43, pp.271-286.

[9] Pan, Y., He, H., Xu, J. and Feinerman, A., 2017. Study of separation force in constrained surface projection stereolithography. Rapid Prototyping Journal, 23(2), pp.353-361.

[10] He, H., Pan, Y., Feinerman, A. and Xu, J., 2018. Air-Diffusion-Channel Constrained Surface Based Stereolithography for Three-Dimensional Printing of Objects with Wide Solid Cross Sections. Journal of Manufacturing Science and Engineering, 140(6), p.061011.

[11] He, H., Xu, J., Yu, X. and Pan, Y., 2018. Effect of Constrained Surface Texturing on Separation Force in Projection Stereolithography. Journal of Manufacturing Science and Engineering, 140(9), p.091007.

[12] He, H., Pan, Y., Xu, J., Yu, X. and Botton, V., 2016. Effect of Surface Texturing on Separation Force in Projection Stereolithography. In 11th International Conference on Micro Manufacturing (ICOMM 2016), Irvine, CA, Mar (pp. 29-31).

[13] He, H., Pan, Y., Xu, J., and Yu, X., 2017, "Effect of Constrained Surface Texturing on Separation Force in Projection Stereolithography," Solid Freeform Fabrication Symposium (SFF), Austin, TX, Aug. 13–15, (pp. 1735–1749).

[14] http://scikit-learn.org/stable/modules/ensemble.html

[15] Yang, Y., Zheng, L., Zhang, J., Cui, Q., Li, Z. and Yu, P.S., 2018. TI-CNN: Convolutional Neural Networks for Fake News Detection. arXiv preprint arXiv:1806.00749.

[16] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. and Shah, R., 1994. Signature verification using a" siamese" time delay neural network. In Advances in neural information processing systems (pp. 737-744).

[17] Hadsell, R., Chopra, S. and LeCun, Y., 2006, June. Dimensionality reduction by learning an invariant mapping. In null (pp. 1735-1742). IEEE.

[18] Koch, G., Zemel, R. and Salakhutdinov, R., 2015. Siamese neural networks for one-shot image recognition. In ICML Deep Learning Workshop (Vol. 2).

[19] Zheng, L., Duffner, S., Idrissi, K., Garcia, C. and Baskurt, A., 2016. Siamese multi-layer perceptrons for dimensionality reduction and face identification. Multimedia Tools and Applications, 75(9), pp.5055-5073.