A real-time iterative machine learning approach for temperature profile prediction in additive manufacturing processes

Arindam Paul*, Mojtaba Mozaffar†, Zijiang Yang*, Wei-keng Liao*, Alok Choudhary*, Jian Cao†, Ankit Agrawal*
*Department of Electrical and Computer Engineering
{arindam.paul,zijiang,wkliao,choudhar,ankitag}@eecs.northwestern.edu
†Department of Mechanical Engineering
{mojtaba, jcao}@mech.northwestern.edu

***Abstract*—Additive Manufacturing (AM) is a manufacturing paradigm that builds three-dimensional objects from a computer-aided design model by successively adding material layer by layer. AM has become very popular in the past decade due to its utility for fast prototyping such as 3D printing as well as manufacturing functional parts with complex geometries using processes such as laser metal deposition that would be difficult to create using traditional machining. As the process for creating an intricate part for an expensive metal such as Titanium is prohibitive with respect to cost, computational models are used to simulate the behavior of AM processes before the experimental run. However, as the simulations are computationally costly and time-consuming for predicting multiscale multi-physics phenomena in AM, physics-informed data-driven machine-learning systems for predicting the behavior of AM processes are immensely beneficial. Such models accelerate not only multiscale simulation tools but also empower real-time control systems using in-situ data. In this paper, we design and develop essential components of a scientific framework for developing a data-driven model-based real-time control system. Finite element methods are employed for solving time-dependent heat equations and developing the database. The proposed framework uses extremely randomized trees - an ensemble of bagged decision trees as the regression algorithm iteratively using temperatures of prior voxels and laser information as inputs to predict temperatures of subsequent voxels. The models achieve mean absolute percentage errors below 1% for predicting temperature profiles for AM processes. The code is made available for the research community at https://anonymous.4open.science/r/112b41b9-05cb-478c-8a07-9366770ee504.**

***Index Terms*—ensemble learning, additive manufacturing, spatio-temporal modeling**

I. Introduction

Additive Manufacturing (AM) is a modern manufacturing approach in which digital 3D design data is used to build parts by sequentially depositing layers of materials [1]. AM techniques are becoming very popular compared to traditional approaches because of their success in building complicated designs, fast prototyping, and low-volume or one-of-a-kind productions across many industries. Direct Metal Deposition (DMD) [2] is an AM technology where various materials such as steel or Titanium are used to develop the finished product. Computational simulations are an essential part of the AM design and optimization as they eliminate the trial and error on expensive manufacturing processes. Finite element-based multi-physics simulation models (FEM) [3], [4] are designed to replicate the AM process before generating the required part using AM. However, FEM-based simulations are computationally costly and time-consuming. This leads to the motivation to develop a predictive tool based on machine learning (ML) that can instantaneously yield the simulation result instead of performing expensive physics-based simulations.

A real-time AM control system can be useful in manufacturing because it can control machines considering the changes in the environment and the machine itself. This can be more important in AM since most of the vital parameters in the quality of final product change considerably during the build process. The temperature field created while building a part using AM is one of the critical components in determining microstructure, porosity and grain size. This system requires a fast data-driven predictive model that can relate machine parameters and replicate desired property behavior accurately using ML techniques, without the need for computationally expensive calculations. There has been an upsurge of interest in the manufacturing community to connect and share data between geographically distributed facilities [5], [6]. We believe a significant amount of experimental data will be available in the near future for manufacturing processes especially AM. This urges the scientific community to develop suitable data-driven tools and techniques.

In this work, we use Generalized Analysis for Multiscale Multi-Physics Application (GAMMA) [7], [8], a FEM based method for developing the database to train our model-based control system. GAMMA is used to solve the time-dependent heat equation and simulate the manufacturing DMD process at the part scale. As the AM process is a spatiotemporal phenomenon (since there is cooling and reheating depending on whether and when a neighboring element is created), any approach for predicting the temperature profile must include the information about neighboring voxels as well as temporal information. In our proposed approach, we harness this characteristic of the AM process during feature reconstruction

for our learning system. The input features for our proposed system include the distance of a given voxel from the current laser beam in the x, y and z axes, laser intensity, time at which the point is created, the time elapsed, and tool speed. One of the advantages of a real-time system is instead of training a prior model ahead of time, one can be developed in-situ. This is crucial for the versatility of ML-driven control system especially as factors such as laser path, laser speed, and laser temperature can largely influence the temperature profile. The proposed approach uses extremely randomized trees (ERTs) [9], an ensemble of bagged decision trees as the regression algorithm, iteratively to develop a model-based control system. A model is developed on the features of first m voxels to predict the temperature of next n voxels at the first stage, and then iteratively a new model is developed at every subsequent stage using the ground-truth temperature of m voxels as well as the predicted temperature of the n voxels.

The result of this work is a real-time iterative supervised predictive model that achieves mean absolute percentage errors (% MAE) below 1% for predicting temperature profiles for AM processes. This staged iterative model outperforms a traditional model that does not use predicted intermediate voxel temperatures. The code is made available for the research community at https://anonymous.4open.science/r/112b41b9-05cb-478c-8a07-9366770ee504/.

The rest of the paper is organized as follows. Section II provides a brief background of AM and DMD, and the FEM code used for developing the database and some related works for application of machine learning in materials informatics, and specifically AM. In Section III, we explain the generation and transformation of the dataset and describe the input features and voxel categories. We describe the motivation and methodology and development of the dataset in Section IV. We discuss the experimental settings and results in Section V, and finally in Section VI, we summarize our conclusions with some future directions.

II. Background and Related Works

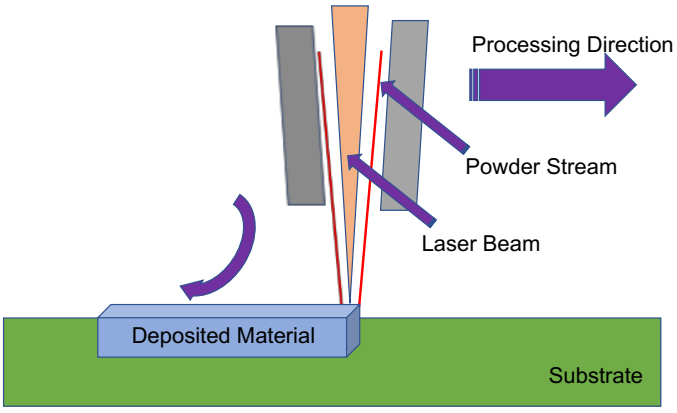
In this section, we present a background of AM and DMD, and the FEM code used for developing the database and some related works for application of machine learning in materials informatics.

A. Additive Manufacturing: Overview

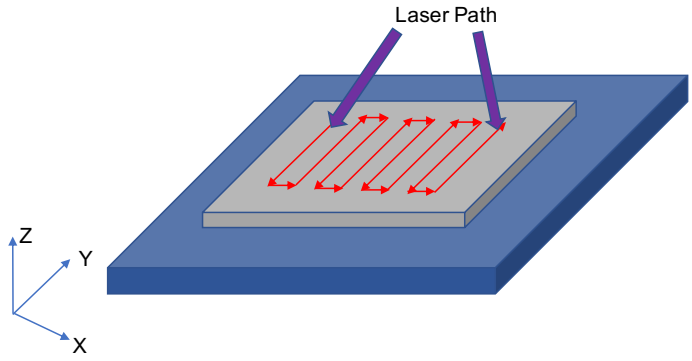
The initial development of a process for creating a three-dimensional object using computer-aided design (CAD) performs a layer by layer deposition was realized due to a desire for rapid prototyping [10], [11]. This was done to reduce the time-cycle of realizing an initial prototype after the conception of design by engineers [12]. Among the major advances that this process presented to product development are the time and cost reduction, and the shortening of the product development cycle. Further, it led to the possibility of creating shapes that were difficult to be machined using traditional manufacturing processes.

AM can appreciably reduce material waste, decrease the amount of inventory, and reduce the number of distinct parts needed for an assembly [13], [14]. Further, AM can reduce the number of steps in a production process, both in the case of tool making as well as direct manufacturing, reducing the need for manual assembly [15]. Besides, AM processes can significantly reduce the total amount of tooling required and its impact on the cost [16]. AM parts can be manufactured in an almost final state thus reducing the amount of connecting parts required to put them together and decreasing part count [17].

B. Direct Metal Deposition



(a) DMD overall setup



(b) DMD Laser Path on Substrate

Fig. 1. Additive Manufacturing using Direct Metal Deposition (DMD) process. The laser source provides the heat while the powder stream provides the metal for the deposition. The metal powder gets melted by the heat from the laser beam and deposited on the substrate. The laser scans over the substrate in a zigzag motion.

DMD [18], [19] is an additive manufacturing technology using a laser to melt metallic powder. DMD processes can produce fully dense, functional metal parts [19] directly from CAD data by depositing metal powders using laser melting and a patented closed-loop control system to maintain dimensional accuracy and material integrity [20]. Heat is generated as a focused heat source such as a laser to sufficiently melt the surface of the substrate and creates a melt pool. A focused powder stream provides material for the melt pool using to form a raised portion of the material. The nozzle is moved over

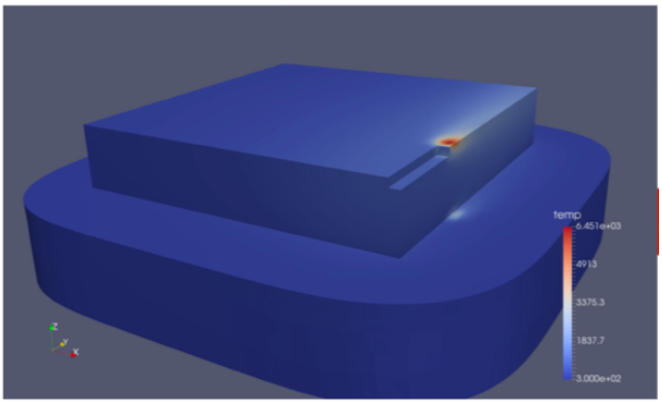


Fig. 2. The simulated metal surface built using DMD is depicted in the figures. The first figure demonstrates the metal created using DMD on the substrate with the temperature scale. The color of the metal surface indicates the spatio-temporal characteristic of the DMD process.

the substrate using a computer-controlled positioning system to create the desired geometry. This is illustrated in Figures 1 and 2 that depict the DMD process and laser motion, and the metal surface built across layers respectively.

C. Finite Element Method Solver

Finite element method (FEM) analysis is a numerical approach for solving differential equations over complex geometries with broad applications in simulating structural properties and fluid dynamics [21]. In this method, first the domain is discretized into small elements, and then a system of equations is assembled over all the elements. GAMMA is an FEM framework that solves transient heat transfer equations for metal powder-based AM processes such as Directed Energy Deposition (DED) and Selective Laser Melting (SLM) [22]. Although an accurate thermal analysis of AM provides vital information for determining microstructure evolution and mechanical performance of the part [4], [23], this kind of analysis can take weeks or months of computing time and therefore too computationally expensive for large-scale problems or optimization purposes [24]. For a given set of processing parameter inputs such as build geometry, laser power, and scan speed, GAMMA calculates spatially-dependent thermal histories within the part, such as temperature profiles and maximum cooling rate. In this work, we use GAMMA to generate the database to train our ensemble model.

D. Related Work

The idle pace of development and deployment of new/improved materials has been deemed as the main bottleneck in the innovation cycles of most emerging technologies [25]. Exploring and harnessing the association between processing, structure, properties, and performance is a critical aspect of new materials exploration [26]–[30]. Data-driven techniques provide faster methods to know important properties of materials and to predict feasibility to synthesize materials experimentally. This can expedite the research process of new materials development. Many initiatives to computationally assist materials discovery using ML techniques have been undertaken [29], [31]–[42].

There has been some limited work on the application of ML techniques for AM processes. Baturynska et al. [43] propounded a conceptual framework for combining FEM and ML methods for optimization of process parameters for powder bed fusion AM. Choy et al. [44] designed a novel recurrent neural network architecture 3D recurrent reconstruction neural network (3D-R2N2) that learned a mapping from images of objects to their underlying shapes in an AM simulation environment. Scime et al. [45]–[47] developed supervised as well as unsupervised models for detecting irregularities and flaws on the laser bed during the AM process.

III. Data

In this section, we explore the generation of the FEM dataset, the transformation of the FEM dataset for machine learning and description of input features and voxel categories.

A. Data Generation

The database for training the supervised model was developed using GAMMA by solving time-dependent heat equations and simulating the manufacturing process at the part scale. It provides temperature and heat flux for every time step for every element that is created during the AM process. In this work, we utilize a GAMMA FEM simulation of 20 mm x 20 mm x 3 mm cuboidal dimensions. A mesh voxel size of edge length 0.5 mm was used. This refers to a cross-section of 40 x 40 voxels along the x (lateral) and y (longitudinal) axes, and Therefore, there are 40 x 40 x 6 voxels in the simulation or 9600 voxels. The time taken for the FEM simulation is about an hour.

The voxel edge length of 0.5 mm chosen in this work is fairly coarse. However, the time taken for a simulation exponentially increases as the mesh voxel size is made finer. For instance, if we reduce the mesh size to half or 0.25 mm, the FEM simulation would take 9 hours. Moreover, the number of data points is in the order of O(n^2) in terms of the voxels. This is because the FEM simulation contains the temperature history of a voxel from the time of creation to the end of the simulation. Therefore, if one voxel is created at each time-step, there will be n data points pertaining to the first voxel created, $n-1$ data points for the second voxels and so on resulting in $n*(n+1)/2$ data points. However, as the laser deposition process creates multiple voxels at the same time-instant, the number of total data points is significantly smaller but nonetheless of the order of O(n^2). This is because each data point corresponds to a unique (x, y, z, t) where (x, y, z) represents an individual voxel and t represents the timestep. In this case, for the 9600 voxels, there are about 9.051652e+06 or about 9.05 million data points. It must be noted that each time-step does not create the same number of voxels as the simulation mimics the weaving (zigzag) motion of the laser (illustrated in Figure 1(b)). More voxels are created during

the lateral movements as compared to when the laser motion reverses.

We chose this simulation size by making a trade-off between a very large simulation that would take days or weeks and potentially create trillions of data points and a small simulation that have too few data points to rigorously train and evaluate the proposed approach.

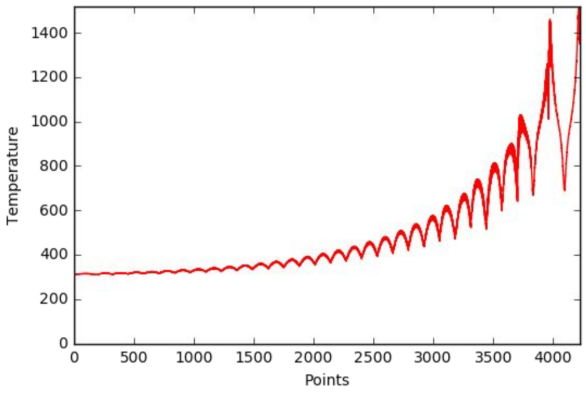
B. Data Pre-processing

Figure 3(a) illustrates the overall temperature profile for the DMD process at the end of the AM process. The index of the point in the x-axis demonstrates the time of creation of the point. We can observe that the points with the lower index or those created earlier slowly approach the room temperature. However, the temperature of the points created later is much higher. Although the overall temperature curve goes higher, we can observe troughs and crests. The troughs are a result of slow cooling of a point created by DMD, and the crest happens when a nearby voxel gets created or heated up. Figure 3(b) illustrates the temperature pattern across different layers are similar as well as across different laser intensities. Therefore, for a higher laser intensity, we can observe a steeper curve. The temperature curves indicate that the AM temperature profile has spatiotemporal as well as other factors dependent on the laser.

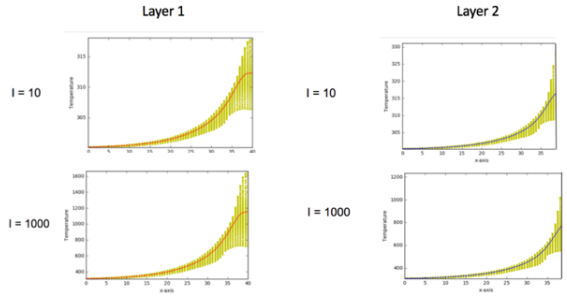
There are many features that impact the temperature of a given voxel. The most important elements are the position of the voxel (x,y,z) and the time elapsed after creation of a voxel. Instead of considering the absolute voxel (x,y,z) position, we consider the distance in the x,y,z with the current position of the laser. The temperature of a given voxel changes with time: cooling or heating. As time passes, the temperature of a given voxel reduces. However, if a new voxel is created proximal to the given voxel, this leads to the increase of the temperature of the voxel. However, the temperature profile fluctuates because of cooling and subsequent reheating due to new material creation. Hence, the feature set for training the supervised model that is agnostic of the temperature of adjacent elements would not provide sufficient information for a supervised learning algorithm to learn the AM process. The temperature of each element is influenced by the temperature of its neighboring elements. The following are the input features used for building the proposed predictive model:

- Historical Features: Temperature of the given voxel at $t-1$ through $t-5$ (if applicable)
- Spatio-Temporal Features: Temperature of neighboring 26 voxels at $t-1$
- Spatial Features: relative x, y and z coordinates of the current voxel with respect to the current position of the laser
- Temporal Features: Time of voxel creation and time elapsed since the creation of given voxel

It is to be noted that the current position of the laser is dependent on both the tool path as well as the tool speed of the laser. Further, it is not necessary that all the input features are available for all the data points. This is possible in case



(a) Temperature Profile of overall Additive Manufacturing Process at the end of the FEM simulation



(b) Temperature Profile across different laser intensities and layers

Fig. 3. Temperature profiles for the DMD process. The temperatures are in Kelvin (K) scale.

of voxels at the edge that does not have neighboring voxels or the absence of temperature history of the given voxel. If the temperature of any feature is missing, we assign a dummy value of -99 as most machine learning algorithms do not accept missing values. One of the essential elements of selecting features is selecting independent attributes. We attempt to build a predictive model which only depends on elements which can be reproducible independent of the dataset on which it has been trained.

C. Voxel Categories

We classify the voxels into 5 categories based on the spatial location of the voxel. As the temperature profiles of voxels surrounded by other voxels may differ from voxels at the periphery, we wanted to investigate if the voxels at the outer edge that have one or more missing neighboring voxels are predicted worse than the interior voxels. This is because our proposed model is dependent on the temperature of the neighboring voxels. To characterize this, we categorize the voxels into 5 categories.

- Interior: all neighboring voxels present
- Edge (Lateral) : neighboring voxel on the x-axis missing
- Edge (Longitudinal) : neighboring voxel on the y-axis missing
- Edge (Vertical) : neighboring voxel on the z-axis missing

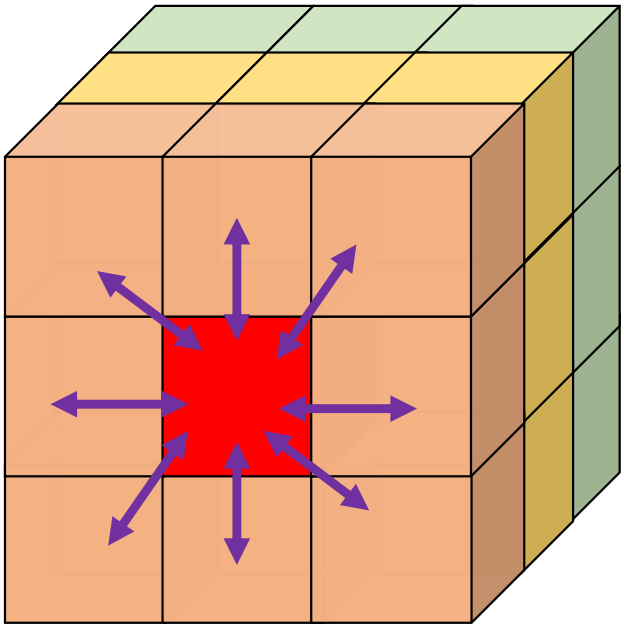


Fig. 4. Illustration of cross-section of the surface to represent conduction of heat on target voxel (labeled in red) from neighboring voxels. However, as this is a 2D cross-section of a voxel, there are 8 neighboring voxels indicated by arrow heads. In three dimensions, a voxel is surrounded by 26 neighboring voxels. The arrows are double-headed as the The different colors of adjacent layers indicate the relative temperature. Layers farther away from a newly created voxel are comparatively cooler : green indicating cool, yellow indicating warm and orange indicating hot.

- Edge (Diagonal) : a neighboring voxel on the planar or cubical diagonal is missing (but no lateral, longitudinal or vertical neighbors are missing)

To avoid confusion, we avoid categorizing a single voxel into multiple categories. If a voxel has a missing neighbor on the x-axis, it is considered an edge (lateral) voxel even if it has a missing y or z-axis neighbor. Similarly, if a voxel has a missing neighbor on y-axis but no missing edge on the x-axis, it is considered as edge (longitudinal) even if there is a missing z-axis neighbor. We decide in this fashion as we can anticipate that newly created voxels might have a missing voxel vertically above (z axis). Therefore, a voxel that has x or y-axis neighbors missing are considered more distinct than a missing z-axis neighbor. If a voxel has any neighbor missing apart from the immediate adjacent neighbor along the x, y and z axes, it is considered a diagonal edge voxel. It is noteworthy that when we categorize a voxel, we do it at a certain time t. This is because for a given newly created voxel at layer l would be an edge(lateral) voxel at the time of creation but the layer $l+1$ is deposited on top of this voxel, it would be an interior voxel.

IV. Method

The motivation and methodology of the proposed iterative approach are outlined in this section. Figure 5 illustrates the flow diagram of the proposed methodology.

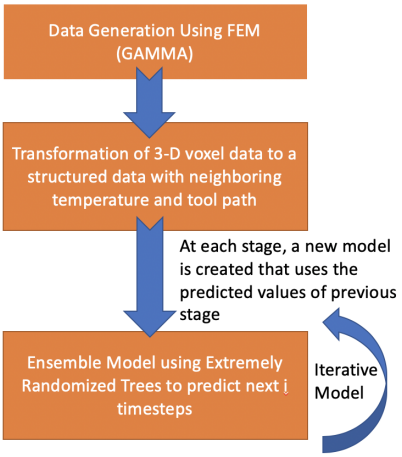


Fig. 5. The overall methodology of the proposed multi-stage iterative model for predicting temperature profile of an additive process

A. Motivation for real-time system

Control systems in manufacturing can be divided into two broad categories [48]. The first class would be error-based control systems in which the changing parameters (parameters of manufacturing machine such as laser power, speed) will be estimated and based on the error that we get from the experiment, we correct our guess until we achieve our desired criteria. The second class would be model-based in which instead of estimating the initial value of machine parameters, they will be selected based on a model which will give us a fairly close answer to the final values.

While an error based control system can be useful in many applications such as motion control, its application in additive manufacturing process parameter control is not common because a significant deviation from the proper value will ruin the part. Developing control manufacturing processes in a way to achieve desired properties in the final product is not a new attempt. It started from simple trial and errors and gradually developed to complicated multiple layer feedback control systems to manipulate system settings for real-time control. However, growing demand for controlling more and more detailed and complicated properties of products overpassed current science and many scientists tried to come up with new methods to overcome this challenge. As a data-driven methodology is more intuitive with a model-based system, our proposed approach outlines such a control system.

B. Iterative Ensemble Model

We explored across many regression algorithms for the developing our models including linear regression (ordinary least square), regularized linear regression : Lasso (L1-regularization) and Ridge (L2-regularization), boosted and bagged decision trees. We did not consider neural networks for this framework. Although, a recurrent neural network model trained on temporal features can be combined with a feed forward neural network trained on non-temporal features, training deep neural networks would take hours to train which

TABLE I
COMPARISON OF PERFORMANCE FOR DIFFERENT MACHINE LEARNING ALGORITHMS WITH CORRESPONDING R^2 AND % MEAN ABSOLUTE ERROR (MAE) BASED ON TRAINING ON THE FIRST 200 TIMESTEPS AND PREDICTING NEXT 300 TIMESTEPS. FOR EACH ALGORITHM, WE EXPLORE VARIOUS HYPERPARAMETERS AND PRESENT THE BEST MODEL.

Algorithm	R^2	**% MAE**	**Training Time (in seconds)**
Linear Regression	0.23	25.08	0.52
Lasso Regression	0.21	23.11	0.53
Ridge Regression	0.38	17.28	0.56
ARIMA	0.15	29.39	0.67
Decision Trees	0.76	9.74	2.30
AdaBoost (20 trees)	0.89	9.40	9.89
AdaBoost (50 trees)	0.92	6.45	55.27
AdaBoost (200 trees)	0.94	3.21	202.58
XGBoost (20 trees)	0.71	13.25	15.65
XGBoost (50 trees)	0.96	2.59	30.92
XGBoost (200 trees)	0.97	2.01	105.67
Random Forest (20 trees)	0.96	1.66	9.88
Random Forest (50 trees)	0.97	1.44	26.68
Extra Trees (20 trees)	**0.99**	**0.81**	7.25
Extra Trees (50 trees)	**0.99**	**0.21**	21.32

is many order of magnitudes time more than the simulation time for FEMs and not feasible for a real-time prediction system where training has happened in-situ. Further, algorithms based on auto regression and moving average such as ARIMA [49] would not be able to capture spatial non-temporal relationships. This is also evident from our benchmarking experiment in Table I.

Algorithms using an ensemble of decision trees have achieved state of the art results for various machine learning tasks [50]. As a non-parametric method like decision trees performed better than parametric methods like linear regression, we decided to explore both boosting and bagging decision trees. Ensemble-based methods have been successful in tackling problems with sequential components [51], [52]. While AdaBoost and XGBoost are algorithms that use boosting in which each successive tree harnesses the decision made by the previous tree, bagged algorithms like Random Forest(RFs) and ERTs make a decision based on the average of many different trees. For both boosting and bagging, weak learners are utilized in form of trees with limited depth. Unlike boosting models which are sequential learners and hence cannot harness running the weak learners in parallel, bagged models can run as many models in parallel as the number of processors. As time of training is essential for a real time application, we choose bagged decision trees and in particular, ERTs as they outperform RFs for our experiments. Table I demonstrate the performance of all the different algorithms when they were trained on first 200 time steps for predicting the next 300 time steps.

ERTs use an ensemble of decision trees in which a node split is selected completely randomly with respect to both variable index and variable splitting value. ERTs are very good generalized learners and perform better in the presence of noisy features. As compared to RFs, ERTs decrease the variance and increase the bias by randomly selecting a node split independent of the splitting value. Both RFs and ERTs perform bootstrap aggregation wherein each weak learner builds a model based on a random sample of observations from the training data, with replacement. Bootstrap aggregation helps in reducing variance in bagged ensembles.

Researchers have proposed rolling recursive or iterative auto regressive moving average modeling [53] for time series prediction. In this work, we decided to explore iterative prediction based on ERTs as we have a combination of historical as well as spatiotemporal features. We propose an iterative staged model in which an initial model is first developed based on the ground-truth data. Then, the data points predicted by the initial model is added to the ground-truth data to develop a model for the next stage which predicts the temperature profile of voxels for future time-steps. We iteratively keep predicting future time-steps using predicted temperature profiles from the previous stage alongside ground-truth data. Figure 6 demonstrates the iterative learning process of our proposed model.

V. EXPERIMENTAL RESULTS

In this section, we present the experimental settings and describe the results of the proposed system for predicting temperature profiles in an AM process. All experiments are carried out using NVIDIA DIGITS DevBox with a Core i7-5930K 6 Core 3.5GHz desktop processor, 64GB DDR4 RAM. The python VTK library [54] was used for processing and converting the voxel data. The data preprocessing as well as most of the regression models were implemented using Scikit-Learn [55]. The XGBoost package [56] was utilized for creating the xgboost model. The ARIMA model was trained from the statsmodels package [57].

For the iterative model, we performed extensive grid-search across various sizes of time step intervals and found the best results when the time step interval was equal to 20. For the experiments, we evaluate with different combinations and ratios of train and test splits. It is to be noted that instead of splitting the train and test set based on a fixed fraction, we divided the dataset based on the timesteps. For instance in Table II, we use data points up to 1000, 800, 500 and 300 timesteps for training and then we predict the next 200, 400, 700, and 900 timesteps respectively. For instance, when we use 800 timesteps for training and 400 for the test set, it corresponds to about 4.34 million training data points and 4.71 million test data points.

Table III compares the timing and regression metrics for the proposed iterative model with a standard non-iterative model that directly predicts temperatures of future time steps varying between 200 to 1000. This experimental design of selecting training data based on timesteps instead of layers also helps in generalizing the training set-up. For instance, the first 200 timesteps would represent few completed layers and an incomplete layer. As at this time, the GAMMA code would have completed simulation only for that part. The same intuition follows for the timesteps in the test set. By training on different timesteps allows us to generalize the framework

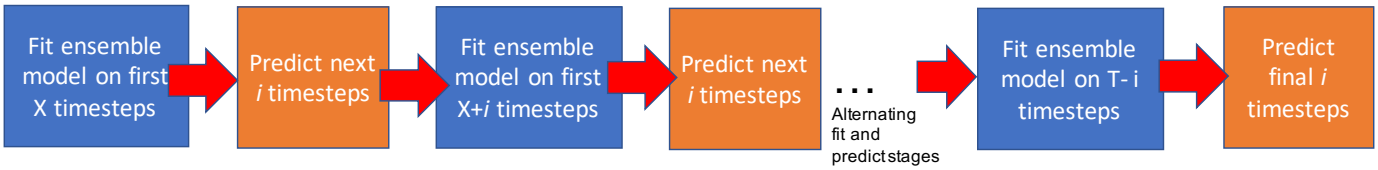


Fig. 6. The proposed staged model using ERTs to predict temperature profiles for additive manufacturing processes. It is to be noted that the number of data-points predicted at each step is not the same as the number of data-points for each voxel. This is because the model predicts not only the temperature of the newly created voxels but also the temperature of the same voxels present in the training set at a later time-step.

TABLE II
COMPARISON OF COMBINATIONS OF TIME-STEPS USED FOR TRAINING AND TEST IN THE STAGED MODEL (WITH CORRESPONDING R^2 AND MEAN ABSOLUTE ERROR%). WE VARY THE NUMBER OF TIME-STEPS USED FOR TRAINING AND VALIDATION. THE TOTAL NUMBER OF TIME-STEPS - SUM OF THE TRAINING AND VALIDATION TIME-STEPS ARE ALWAYS EQUAL TO 1200.

Training		**Test**		R^2	**% MAE**
No. of timesteps	**No. of datapoints (in millions)**	**No. of timesteps**	**No. of datapoints (in millions)**		
1000	6.75	200	2.30	0.992	0.289
800	4.34	400	4.71	0.989	0.679
500	1.72	700	7.33	0.982	1.329
300	0.63	900	8.42	0.972	1.848

TABLE III
COMPARISON OF PROPOSED ITERATIVE MODEL WITH A DIRECT MODEL THAT DIRECTLY PREDICTS THE TEMPERATURE OF SUBSEQUENT POINTS. WE PRESENT THE TIME TAKEN AS WELL AS REGRESSION METRICS (CORRESPONDING R^2 AND MEAN ABSOLUTE ERROR%) FOR BOTH THE MODELS. THE INITIAL NUMBER OF TIME-STEPS USED FOR TRAINING IS SET TO 200 AND THE SIZE OF THE ITERATION IS SET AS 20 TIME-STEPS. WE VARY THE NUMBER OF FUTURE TIME-STEPS PREDICTED.

Iterations	**Future Timesteps Predicted**	**Iterative Model**			**Standard Model**		
		Time (in seconds)	R^2	**% MAE**	Time (in seconds)	R^2	**% MAE**
10	200	68.69	0.989	0.675	0.293	0.921	5.39
20	400	137.08	0.978	1.444	0.308	0.906	5.71
30	600	210.04	0.976	1.489	0.317	0.876	6.07
40	800	278.61	0.971	1.903	0.480	0.861	6.55
50	1000	353.96	0.969	1.721	0.590	0.794	6.63

to different shapes. Although the iterative model performs much better than the direct model, we must stress that the direct model is much faster. For example, while predicting the temperature for 1000 future time steps, the iterative model takes 353.96 seconds, the direct model requires 0.29 seconds. However, we can observe that the % MAE value of the direct model is much worse as compared to the iterative model. While the iterative model has R^2 between 0.97 and 0.99 and % MAE between 0.68 to 1.73 %, the direct model has R^2 between 0.79 and 0.92 and % MAE between 5.39 to 6.63 %.

The results in Table IV illustrate that interior and edge (vertical) voxels comprise the bulk of the voxels (40.15% and 49.20%). This is anticipated as for any new layer created, none of the voxels in the new layer would have a vertical neighbor until a new layer is deposited. We also find that there is no significant difference in the prediction accuracy between the type of voxels. This demonstrates further that our iterative prediction model is able to learn the temperature profiles for both edge voxels as well as interior voxels.

TABLE IV
COMPARISON OF R^2 AND MEAN ABSOLUTE ERROR% ACROSS THE DIFFERENT TYPES OF VOXEL

Type of voxel	**% of overall voxels**	R^2	**% MAE**
Interior	40.15	0.990	0.916
Edge (Lateral)	4.92	0.992	0.898
Edge (Longitudinal)	5.09	0.988	0.923
Edge (Vertical)	49.20	0.989	0.918
Edge (Diagonal)	0.63	0.988	0.926

Table V depict how varying the number of estimators (trees, in the case of ERTs) impacts the overall time (sum of the training and prediction times). As expected, the % MAE reduces and R^2 increases as the number of estimators increase. The variance of bagged ensembles reduce as more trees are used to make the prediction, and MAE reduces with variance. However, as the overall time increases with the number of

TABLE V
COMPARISON OF NUMBER OF TREES/ESTIMATORS IN THE ENSEMBLE. AS WE VARY THE NUMBER OF ESTIMATORS, WE PRESENT THE TRADE-OFF IN THE FORM OF TIME AND R^2 AND MEAN ABSOLUTE ERROR%. THE NUMBER OF VOXELS PREDICTED IN EACH ITERATION IS 25, AND THERE ARE 40 STEPS IN EACH ITERATION

No. of estimators	**Overall Time** (in seconds)	R^2	**% MAE**
4	154.5	0.964	2.14
10	257.5	0.970	1.38
20	493.2	0.975	1.29
50	902.4	0.981	1.03

estimators, any deployed system would need to make a trade-off between reducing the % MAE and the cost and time of the available computing resources.

Figure 7 illustrates the impact of the temperature profiles of the voxels immediately surrounding the target voxel for which we are predicting the temperature profile. The voxels on the x-axis have a more significant impact than the voxels on the y-axis. This is expected as the direction of the laser is towards the x-axis. Further, the importance of the $T\_immediate\_(y+1)$ and $T\_immediate\_(y-1)$ features are equal and this is also unsurprising as the laser path zig-zags on the y-axis during the AM process (as illustrated in 2b) and is therefore agnostic of the directionality in the y-axis. Figure 8 depicts the scatterplot for the predicted vs. the ground-truth FEM voxel temperatures. We can observe that the prediction accuracy increases with the number of estimators. Further, we have fewer outliers when the number of estimators is higher. This is expected as bagged ensembles perform well based on crowd-sourcing the prediction of weak learners which are likely to have high bias on their own but have low bias overall as an ensemble.

The primary motivation of this work was to develop a ML-aided framework that can reduce or replace FEM simulations. Hence, it was very important to have a model that has a low MAE guarantee. ERTs are especially effective at creating data-driven rules for handling different kinds of data points. For a voxel that has been created long ago such as in the first layer, the temperature of the voxel would not change as a new voxel is created at the topmost layer. However, the temperature of a given voxel created few time steps or a voxel created many time steps before but immediately below a newly created voxel would be high. Not only are ERTs fast to train, they are easy to interpret as we can rank the features as well as visualize the different candidate trees. Interpretability of algorithms is extremely important in the scientific and engineering community.

VI. CONCLUSIONS AND FUTURE WORK

This paper presents essential components of a scientific framework for a model-based real-time AM control system. The proposed approach utilizes extremely randomized trees - an ensemble of bagged decision trees as the regression algorithm iteratively using temperatures of prior voxels and laser information as inputs to predict temperatures of subsequent voxels, and is able to achieve % MAE below 1% for predicting temperature profiles. One of the advantages of a real-time system is instead of training a prior model ahead of time, one can be trained in-situ. It is crucial for the versatility of the AM ML-driven simulation process especially as factors such as laser path, laser speed, and laser temperature can largely influence the temperature profile.

In future, we plan to explore the impact of voxel mesh size on the prediction results across coarse to finer mesh. The next goal of this framework is to be part of an interleaved FEM-ML control system that harnesses the temperature profile of the odd layer (Layer i) calculated using FEM to predict the subsequent even layer (Layer $i+1$). Layer $i+2$ will then be calculated using FEM simulation, and Layer $i+3$ will be predicted. This can accelerate the speed of simulations by nearly a factor of two, hopefully without impacting the accuracy significantly.

ACKNOWLEDGMENT

This work was performed under the following financial assistance award 70NANB14H012 from U.S. Department of Commerce, National Institute of Standards and Technology as part of the Center for Hierarchical Materials Design (CHiMaD) as primary support. In addition, authors also acknowledge partial support from the following grants: AFOSR award FA9550-12-1-0458; DARPA award N66001-15-C-4036; NSF award CCF-1409601; DOE awards DE-SC0007456, DE-SC0014330.

REFERENCES

[1] JK Watson and KMB Taminger. A decision-support model for selecting additive manufacturing versus subtractive manufacturing based on energy consumption. *Journal of Cleaner Production*, 176:1316–1322, 2018.

[2] Yaoyu Ding, James Warton, and Radovan Kovacevic. Development of sensing and control system for robotized laser-based direct metal addition system. *Additive Manufacturing*, 10:24–35, 2016.

[3] J Ding, P Colegrove, Jorn Mehnen, Supriyo Ganguly, PM Sequeira Almeida, F Wang, and S Williams. Thermo-mechanical analysis of wire and arc additive layer manufacturing process on large multi-layer parts. *Computational Materials Science*, 50(12):3315–3322, 2011.

[4] Wentao Yan, Stephen Lin, Orion L Kafka, Yanping Lian, Cheng Yu, Zeliang Liu, Jinhui Yan, Sarah Wolff, Hao Wu, Ebot Ndip-Agbor, et al. Data-driven multi-scale multi-physics models to derive process–structure–property relationships for additive manufacturing. *Computational Mechanics*, 61(5):521–541, 2018.

[5] Jorge E Correa, Ricardo Toro, and Placid M Ferreira. A new paradigm for organizing networks of computer numerical control manufacturing resources in cloud manufacturing. *Procedia Manufacturing*, 26:1318–1329, 2018.

[6] Daniel J Garcia, Mojtaba Mozaffar, Huaqing Ren, Jorge E Correa, Kornel Ehmann, Jian Cao, and Fengqi You. Sustainable manufacturing with cyber-physical discrete manufacturing networks: Overview and modeling framework. *Journal of Manufacturing Science and Engineering*, 141(2):021013, 2019.

[7] Jacob Smith, Wei Xiong, Jian Cao, and Wing Kam Liu. Thermodynamically consistent microstructure prediction of additively manufactured materials. *Computational mechanics*, 57(3):359–370, 2016.

[8] Mojtaba Mozaffar, Ebot Ndip-Agbor, Stephen Lin, Gregory J Wagner, Kornel Ehmann, and Jian Cao. Acceleration strategies for explicit finite element analysis of metal powder-based additive manufacturing processes using graphical processing units. *Computational Mechanics*, pages 1–16, 2019.

[9] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

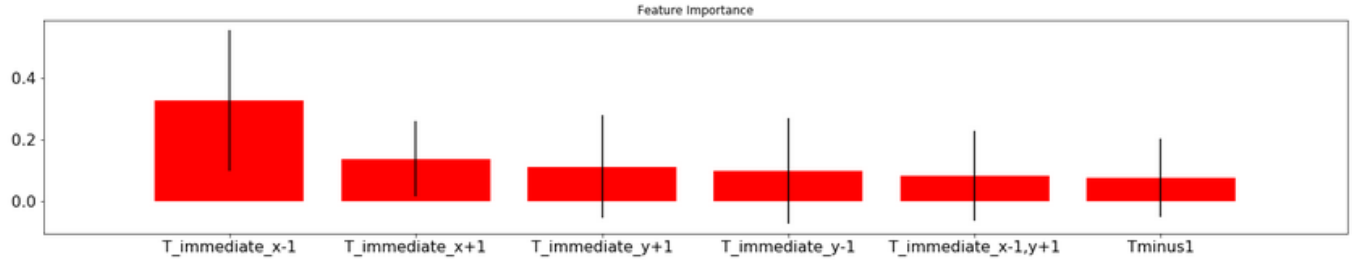
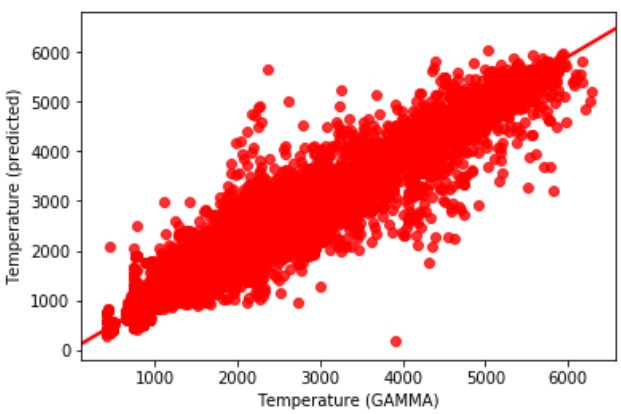
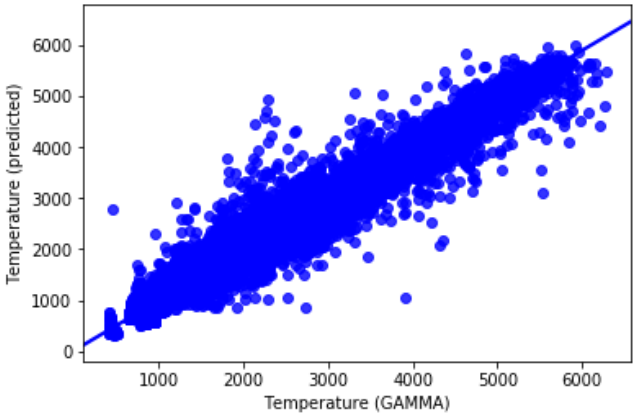


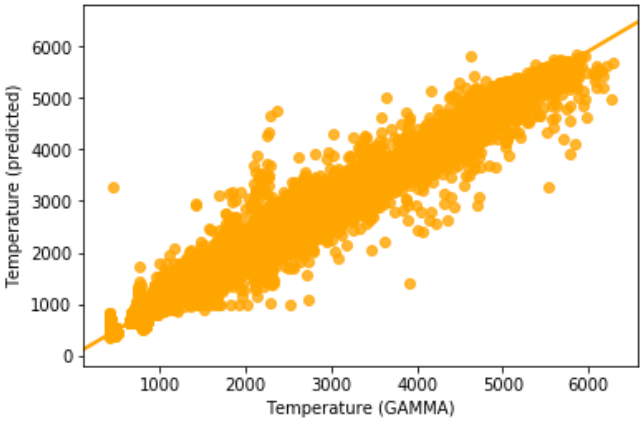
Fig. 7. The feature importance for the top input features in the ensemble semi-supervised approach



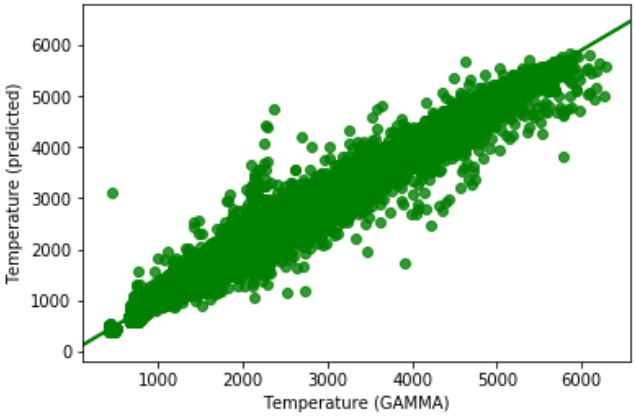
(a) 4 trees



(b) 10 trees



(c) 20 trees



(d) 50 trees

Fig. 8. Scatterplot for predicted vs. FEM temperatures. As the number of estimators/trees increase, the prediction accuracy improves.

[10] William E Frazier. Metal additive manufacturing: a review. *Journal of Materials Engineering and Performance*, 23(6):1917–1928, 2014.

[11] Kaufui V Wong and Aldo Hernandez. A review of additive manufacturing. *ISRN Mechanical Engineering*, 2012, 2012.

[12] Tuan D Ngo, Alireza Kashani, Gabriele Imbalzano, Kate TQ Nguyen, and David Hui. Additive manufacturing (3d printing): A review of materials, methods, applications and challenges. *Composites Part B: Engineering*, 2018.

[13] Bernhard Mueller. Additive manufacturing technologies–rapid prototyping to direct digital manufacturing. *Assembly Automation*, 32(2), 2012.

[14] J-P Kruth, Ming-Chuan Leu, and Terunaga Nakagawa. Progress in additive manufacturing and rapid prototyping. *Cirp Annals*, 47(2):525–540, 1998.

[15] Jeremy Faludi, Cindy Bayley, Suraj Bhogal, and Myles Iribarne. Comparing environmental impacts of additive manufacturing vs traditional machining via life-cycle assessment. *Rapid Prototyping Journal*, 21(1):14–33, 2015.

[16] Ville Matilainen, Heidi Piili, Antti Salminen, Tatu Syvänen, and Olli Nyrhilä. Characterization of process efficiency improvement in laser additive manufacturing. *Physics Procedia*, 56:317–326, 2014.

[17] Samuel H Huang, Peng Liu, Abhiram Mokasdar, and Liang Hou. Additive manufacturing and its societal impact: a literature review. *The International Journal of Advanced Manufacturing Technology*, 67(5-8):1191–1203, 2013.

[18] Gary K Lewis and Eric Schlienger. Practical considerations and capabilities for laser assisted direct metal deposition. *Materials &*

Design, 21(4):417–423, 2000.

[19] P Peyre, P Aubry, R Fabbro, R Neveu, and Arnaud Longuet. Analytical and numerical modelling of the direct metal deposition laser process. *Journal of Physics D: Applied Physics*, 41(2):025403, 2008.

[20] Huan Qi, Jyotirmoy Mazumder, and Hyungson Ki. Numerical simulation of heat transfer and fluid flow in coaxial laser cladding process for direct metal deposition. *Journal of applied physics*, 100(2):024903, 2006.

[21] Jacob Fish and Ted Belytschko. *A first course in finite elements*, volume 1. John Wiley & Sons New York, 2007.

[22] Jacob Smith, Wei Xiong, Wentao Yan, Stephen Lin, Puikei Cheng, Orion L Kafka, Gregory J Wagner, Jian Cao, and Wing Kam Liu. Linking process, structure, property, and performance for metal-based additive manufacturing: computational approaches with experimental support. *Computational Mechanics*, 57(4):583–610, 2016.

[23] Sarah J Wolff, Stephen Lin, Eric J Faierson, Wing Kam Liu, Gregory J Wagner, and Jian Cao. A framework to link localized cooling and properties of directed energy deposition (ded)-processed ti-6al-4v. *Acta Materialia*, 132:106–117, 2017.

[24] Marianne M Francois, Amy Sun, Wayne E King, Neil Jon Henson, Damien Tourret, Ccut Allan Bronkhorst, Neil N Carlson, Christopher Kyle Newman, Terry Scot Haut, Jozsef Bakosi, et al. Modeling of additive manufacturing processes for metals: Challenges and opportunities. *Current Opinion in Solid State and Materials Science*, 21(LA-UR-16-24513), 2017.

[25] Surya R Kalidindi. Data science and cyberinfrastructure: critical enablers for accelerated development of hierarchical materials. *International Materials Reviews*, 60(3):150–168, 2015.

[26] Gregory B Olson. Computational design of hierarchically structured materials. *Science*, 277(5330):1237–1242, 1997.

[27] Ankit Agrawal and Alok Choudhary. Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science. *APL Materials*, 4(5):053208, 2016.

[28] Veera Sundararaghavan and Nicholas Zabaras. Linear analysis of texture–property relationships using process-based representations of rodrigues space. *Acta Materialia*, 55(5):1573–1587, 2007.

[29] Bryce Meredig, Ankit Agrawal, Scott Kirklin, James E Saal, JW Doak, A Thompson, Kunpeng Zhang, Alok Choudhary, and Christopher Wolverton. Combinatorial screening for new materials in unconstrained composition space with machine learning. *Physical Review B*, 89(9):094104, 2014.

[30] Ruijin Cang, Yaopengxiao Xu, Shaohua Chen, Yongming Liu, Yang Jiao, and Max Yi Ren. Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design. *Journal of Mechanical Design*, 139(7):071404, 2017.

[31] Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials*, 2:16028, 2016.

[32] Zachary D Pozun, Katja Hansen, Daniel Sheppard, Matthias Rupp, Klaus-Robert Müller, and Graeme Henkelman. Optimizing transition states via kernel-based machine learning. *The Journal of chemical physics*, 136(17):174101, 2012.

[33] Jaimyun Jung, Jae Ik Yoon, Hyung Keun Park, Jin You Kim, and Hyoung Seop Kim. An efficient machine learning approach to establish structure-property linkages. *Computational Materials Science*, 156:17–25, 2019.

[34] Dipendra Jha, Logan Ward, Arindam Paul, Wei-keng Liao, Alok Choudhary, Chris Wolverton, and Ankit Agrawal. Elemnet: Deep learning the chemistry of materials from only elemental composition. *Scientific reports*, 8(1):17593, 2018.

[35] Alireza Rahnama, Sam Clark, and Seetharaman Sridhar. Machine learning for predicting occurrence of interphase precipitation in hsla steels. *Computational Materials Science*, 154:169–177, 2018.

[36] Arindam Paul, Dipendra Jha, Reda Al-Bahrani, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Chemixnet: Mixed dnn architectures for predicting chemical properties using multiple molecular representations. In *Proceedings of the Workshop on Molecules and Materials at the 32nd Conference on Neural Information Processing Systems*, 2018.

[37] Aaron Gilad Kusne, Tieren Gao, Apurva Mehta, Liqin Ke, Manh Cuong Nguyen, Kai-Ming Ho, Vladimir Antropov, Cai-Zhuang Wang, Matthew J Kramer, Christian Long, et al. On-the-fly machine-learning for high-throughput experiments: search for rare-earth-free permanent magnets. *Scientific reports*, 4, 2014.

[38] Arindam Paul, Pinar Acar, Wei-keng Liao, Alok Choudhary, Veera Sundararaghavan, and Ankit Agrawal. Microstructure optimization with constrained design objectives using machine learning-based feedback-aware data-generation. *Computational Materials Science*, 160:334–351, 2019.

[39] Kazumasa Tsutsui, Hidenori Terasaki, Tatsuya Maemura, Kotaro Hayashi, Koji Moriguchi, and Shigekazu Morito. Microstructural diagram for steel based on crystallography with machine learning. *Computational Materials Science*, 159:403–411, 2019.

[40] Arindam Paul, Dipendra Jha, Reda Al-Bahrani, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Transfer learning using ensemble neural networks for organic solar cell screening. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019.

[41] Rampi Ramprasad, Rohit Batra, Ghanshyam Pilania, Arun Mannodi-Kanakkithodi, and Chiho Kim. Machine learning in materials informatics: recent applications and prospects. *npj Computational Materials*, 3(1):54, 2017.

[42] Dipendra Jha, Saransh Singh, Reda Al-Bahrani, Wei-keng Liao, Alok Choudhary, Marc De Graef, and Ankit Agrawal. Extracting grain orientations from ebsd patterns of polycrystalline materials using convolutional neural networks. *Microscopy and Microanalysis*, 24(5):497–502, 2018.

[43] Ivanna Baturynska, Oleksandr Semeniuta, and Kristian Martinsen. Optimization of process parameters for powder bed fusion additive manufacturing by combination of machine learning and finite element method: A conceptual framework. *Procedia CIRP*, 67:227–232, 2018.

[44] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016.

[45] Luke Scime and Jack Beuth. Anomaly detection and classification in a laser powder bed additive manufacturing process using a trained computer vision algorithm. *Additive Manufacturing*, 19:114–126, 2018.

[46] Luke Scime and Jack Beuth. A multi-scale convolutional neural network for autonomous anomaly detection and classification in a laser powder bed fusion additive manufacturing process. *Additive Manufacturing*, 24:273–286, 2018.

[47] Luke Scime and Jack Beuth. Using machine learning to identify in-situ melt pool signatures indicative of flaw formation in a laser powder bed fusion additive manufacturing process. *Additive Manufacturing*, 25:151–165, 2019.

[48] Gustavo Tapia and Alaa Elwany. A review on process monitoring and control in metal-based additive manufacturing. *Journal of Manufacturing Science and Engineering*, 136(6):060801, 2014.

[49] Ulrich Helfenstein. *ARMA and ARIMA Models*. American Cancer Society, 2005.

[50] Thomas G Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine learning*, 40(2):139–157, 2000.

[51] Sadi Evren Seker, Cihan Mert, Khaled Al-Naami, Ugur Ayan, and Nuri Ozalp. Ensemble classification over stock market time series and economy news. In *2013 IEEE International Conference on Intelligence and Security Informatics*, pages 272–273. IEEE, 2013.

[52] Manish Kumar and M Thenmozhi. Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper*, 2006.

[53] Víctor M Guerrero and J Martínez. A recursive arima-based procedure for disaggregating a time series variable using concurrent data. *Test*, 4(2):359–376, 1995.

[54] vtk · pypi. https://pypi.org/project/vtk/.

[55] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[56] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, and Yuan Tang. Xgboost: extreme gradient boosting. *R package version 0.4-2*, pages 1–4, 2015.

[57] Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, volume 57, page 61. Scipy, 2010.