Machine Learning Model Towards Evaluating Data gathering methods in Manufacturing and Mechanical Engineering

Mahyar Amini [1,3], Koosha Sharifani[2], Ali Rahmani [3]
[1] University Technology Malaysia (UTM), Malaysia
[2] University of North Carolina at Charlotte, United States
[3] MahamGostar Research Group, Iran

ABSTRACT

Supervised Machine Learning (ML) models require extensive training data to properly approximate the behavior of complex mechanical processes and systems. Real-world experiments or adequate simulations are expensive, time-consuming or incident-related and make the efficient acquisition of sample data a compelling necessity. In mechanical engineering and manufacturing, data is usually collected via established Design of Experiments (DOE) methods. At the same time, the topic of Active Learning (AL) is gaining in importance in the research community and promises a reduction in the amount of data, but is rarely used in industry. In this paper, we compare the most common data sampling methods with AL to achieve better predictive results with fewer samples on regression tasks. We propose a novel evaluation framework that allows to compare various sampling methods in a controlled and unbiased manner, regardless of their different requirements. Using three exemplary use cases (UCs), we evaluate when one should use AL or DOE methods for the task of data generation, by looking at the sample efficiency, stability and predictive accuracy of the resulting ML models. This paper provides practical guidance to both engineers and data scientists, who required highly efficient data collection for later use of ML.

KEYWORDS: additive manufacturing, machine learning, design of experiments, data generation

1.0 INTRODUCTION

Training and deploying ML (Machine Learning) models have become considerably easy because of the available open source frameworks. This results in active usage of advanced ML algorithms across various domains. One of remaining substantial entry barriers to the world of ML remains the data availability. Domain-specific data sets can be extremely difficult to find. Many leading companies such as Google and Facebook concentrate on collecting and monetizing the data [1-5]. The field of mechanical engineering and manufacturing is not an exception, companies strive to improve their product design and manufacturing processes to gain a competitive edge through utilization of available data and advanced analytics tools. Already available ML solutions cover a wide range of real-world applications such as approximating complex systems, speeding up and enhancing simulations topology-optimization and energy management [6-11]. Yet addressing a certain problem often requires the acquisition of training data first, which means running numerous time- and cost-extensive trials on real equipment or simulated environments. Besides, a large quantity of data does not necessarily guarantee the success of the given project, since the data must be available in a certain quality and variability. As it is hard to predict the outcome, related costs and required capacities of a real-world experiment, the acceptance level among manufacturing companies remains limited [12-17]. All of this turns the implementation of ML solutions in the context of complex production systems and processes into a challenging task. It is important to choose an optimal data generation strategy specifically suitable for the intended ML solution to ensure the success of the whole project. For a certain UC (Use Case), it can be challenging to make a well-founded decision upon how to generate required data. Since most AL (Active Learning) studies take random sampling as performance baseline it stays unclear if and how much of the performance gain can AL offer over a traditional well planned DOE [18-24]. To our best knowledge, there is no study available comparing directly DOE methods with AL within a context of mechanical engineering and manufacturing. In this paper, we address these research gap by directly comparing the random generation of data, Latin Hypercube Sampling (LHS), which is a common DOE method, and Query-by-Committee (QBC) sampling from the area of AL. To do so we, develop a novel evaluation framework allowing for direct comparison of DOE and AL methods in a controlled and bias-free manner [25-33]. We ensure the relevancy of the results specifically for

mechanical engineering and manufacturing tasks by using appropriate UCs of different complexity. The suitability of all data generation approaches is evaluated in terms of sample efficiency, stability of results and achieved prediction accuracy of ML models trained on the generated data [34-41]. The evaluation is conducted with the help of LGBM (Light Gradient Boosting Machine), based on regression trees, and a method to boost the prediction performance by combining several ML models called bagging. Both algorithms are known and widely-used in ML applications across different domains. The results serve as a practical guide for engineers and data scientists, when selecting data generation strategies for ML projects in the mechanical engineering and production [42-49].

2.0 BACKGROUND

This section summarizes and reviews state of the art data generation approaches and AL methods as a subarea of ML.

A. Data Generation Approaches

Leveraging supervised ML models for predicting complex systems requires large amounts of samples with sufficient variance. Generating a labeled sample requires execution and observation of a real process or a realistic simulation [50-56]. Research in the area of DOE proposes several methods for choosing samples to be labeled, which are widely used due to their simplicity or versatility, and described in the following:

1) Completely randomized designs: Completely randomized designs assign a value for each feature randomly. It is a basic technique and was presented in 1964. Small numbers of samples are not necessarily well-spread, but random sampling achieves good performance for large sample counts. Figure 1 depicts example distributions for random sampling, which is easy to implement and is used as one performance baseline for our evaluation [57-63].

2) Randomized block designs: Randomized block designs are an extension of completely randomized designs. The basic concept is to separate the value space of each feature into homogeneous blocks and randomly sample within these blocks. The chosen samples are distributed uniformly among the defined blocks, which leads to a random-ordered uniform distribution with additional noise [64-71]. A widely-used approach is the Latin Hypercube Sampling (LHS), which combines random sampling's scalability with the homogeneous space- filling of classical designs. Figure 1 depicts example distributions for LHS. If we consider a k-dimensional experimental region D and split every dimension into n uniform blocks, LHS created n cubes of equal probability [72-76].

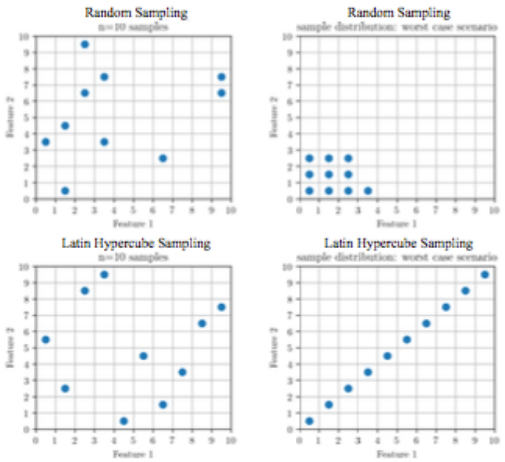


Fig. 1: Two-dimensional random and LHS, n=10.

3) Full factorial designs: Full factorial designs are com- mon sampling designs where all features are

set to a fixed number of l levels each. Popular choices, in engineering at least, are two-level designs (l = 2), where each feature can either take its high or low value. A full factorial design creates all possible combinations of the available values for each of the k features. For n features with l levels each, this leads to l experiment runs. Note that even for seven features with three levels only (low, mid and high), full factorial designs create 2187 samples. The exponential growth makes experiments with more features not suitable, especially at high costs per experiments [77-81].

4) Fractional factorial designs: A fractional factorial de- sign is defined as a "factorial experiment in which only an adequately chosen fraction of the treatment combinations required for the complete factorial experiment is selected to be run". The experiment runs generated by a full factorial design are reduced by only taking a fraction of these [1-7]. The success of the considered method is dependent on the decision of what parts of the experiment to keep. Basic approaches pick fractions such as, etc. and typically achieve good results 24 when the chosen experiment runs are both orthogonal and balanced. For the above full factorial example with n features with l levels each, a fractional design reduces the number of runs by the chosen denominator d to 1 ∗ 1*n [8-13].

5) Response surface designs: According to projects, classical designs like Box-Behnken Design (BBD) [18] and Central Composite Design (CCD) are mainly used for physical experiments, as they select a well spread and small number of samples from the feature space. Figure 2 depicts a sample selection of BBD and CCD for three features. As described in, the number of samples in these classical designs is fixed and directly depends on the number n of features included in the experiment. Given three (n = 3) dimensions, BBD creates 13 samples arranged in a sphere. Except for the centre point, all samples are situated on the edges of the experimental region. CCD create 15 samples for n = 3, which are aligned as a group of "star points" around one center point (cf. Figure 2) [14-19].

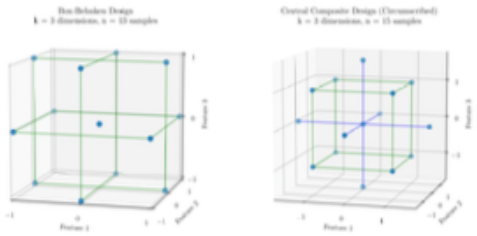


Fig. 2: Left: BBD with one center point. Right: CCD with star points aligned around the center.

B. Active Learning

AL algorithms belong to a sub-field of ML and use in cremental query learning that consists of two components: A query algorithm and a training algorithm, typically an ML model. The main hypothesis is that the training algorithm can learn more with fewer samples, if it can select the samples itself via the query algorithm. Figure 3 depicts a typical AL loop, where the main goal is the optimal training of an ML model with as few samples as possible. The AL algorithm selects one or more samples (step size) from the unlabeled train set U and an oracle returns the respective label for training the ML model(s) further. For n features, the feature space X is the set of all possible feature values combinations: X = {(⃗x) | ⃗x is a valid feature combination}. Note that X is continuous and thus can contain an infinite amount of entries. P ⊂ X is a finite subset that we also call *pool* in the AL context. Further, the labeled train set L is the subset of P where labels are known:L={p∈P| labelofpisknown}Theunlabeled train set U respectively is defined as U = P \ L. The oracle shown in Figure 3 can be a human annotator [20], a real-world process or a simulation (e.g., of a produc- tion process). The main difference from the statistical DOE described in Section II-A is that an AL method extends its train set actively [20-28]. Every next sample included in the train set is expected to maximize the learning effect. This strategy selects that instance from U where the underlying ML model is least certain regarding the prediction. Numerous query algorithms are available, the two most

commonly used and well-known approaches are uncertainty sampling and Query- By-Committee (QBC) [21]. We do not include *uncertainty sampling* in our work, because it is only applicable when a probabilistic ML model is used. The QBC method was introduced in and contains a committee $C = \{\theta_1, ..., \theta_C\}$ that is trained on the labeled data set L. The next sample is chosen based on the *principle of maximal disagreement*. All committee members $\theta_i \in C$ predict a label for every available $x \in U$, and the sample with the highest disagreement in predicted labels is selected. This approach requires the committee members θ_i to have different hypotheses, as explained in Figure 4 [29-34]. QBC focuses on controversial regions in the unlabeled train set U, queries that sample $x \in U$ with the largest disagreement, and improves all members of the committee C by including x while re-fitting these. QBC is reported to lead to consistently good results compared to other approaches [23]. Moreover, QBC is one of the best- documented and implemented AL algorithms as of today with several Python implementations [35-41].

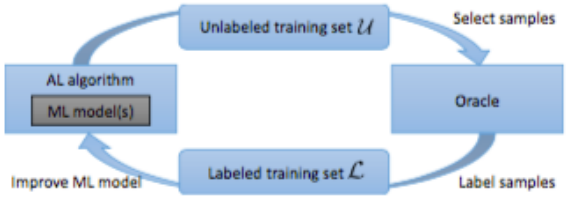


Fig. 3: The AL algorithm repeatedly selects samples, an oracle labels these, and its results improve the ML model(s).

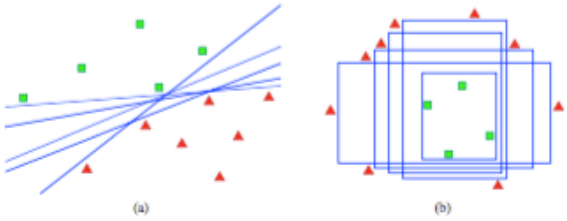


Fig. 4: Different hypotheses of (a) linear and (b) axis-parallel binary classifiers, which are all consistent with the labeled training data $\mathcal{L}$ [20].

3.0 RELATED WORK

A. DOE Applications in Machine Learning

DOE can facilitate the application of many computationally- expensive algorithms. A big portion of the research is dedicated to the use of DOE for algorithm-tuning. Caserta and Voss adopt Response Surface Methodology (RSM) to fine-tune a proposed Corridor Method algorithm on an example of a block relocation task in container terminal logistics. Factorial Experimental Design combined with a local search algorithm can be successfully used for finding parameter values for algorithms whose performance depends on using non-obvious combinations of values. Response Surface Experimental Design was used in to demonstrate the DOE suitability for tuning of optimization algorithms on an ant colony system for the travelling salesman problem example. Several studies show that DOE usage in the parameter-tuning process of ML Algorithms can lead to good results with reasonable computational costs [1-17].

DOE proves to be a suitable vehicle to address various challenges in ML applications from feature selection for predictive model training to behavior investigations of neuronal networks in the context of reinforcement learning. Subsequently, DOE is a well-tailored tool for the generation of training data while creating ML models [18-27].

B. Data Generation in Machine Learning

There are several different approaches to generate data for training of ML Models. Often data is generated using simple random sampling. For instance, Moiz et al. use random sampling methods to generate data for a behavior approximation of heavy-duty engine simulation with ML model. However, it is widely assumed that DOE and active learning approaches may allow for better performance with less training data required for real-life ML applications [28-36].

Such Pfrommer et al. assume a possible reduction of required training data with the help of DOE approaches to train deep neuronal networks for the optimization of composite textile draping process. Probst et al. use optimal Latin hypercube DOE to map input-output parameters of a complex computational fluid dynamics (CFD) simulation and train predictive models on it. To conclude, currently there are various ways to proceed with data generation for subsequent ML model training with little practical guidance on what method to chose under which conditions [37-43].

4.0 DEVELOPING AN EVALUATION FRAMEWORK

The overall goal of our experiments is to compare AL with two classical DOE methods, namely random sampling and LHS. An AL method can only be considered as production- ready, if it consistently outperforms well-designed DOE methods without prior tuning and trial runs [44-49].

TABLE I: Data requirements of the compared data generation strategies and solutions proposed in our evaluation framework.

Strategy	Required input data	Framework solution
Random sampling	Continuous	GTM
LHS	Continuous	GTM
QBC	Discrete and finite	Create pool $\mathcal{P}$ via GTM

A. Input Data Requirements

Table I summarizes the data requirements of random sampling, LHS and QBC. Because real-world data set do not meet these requirements for random sampling or LHS, we develop an evaluation framework that includes solutions to this. This framework also provides a proper data basis for QBC and is described in the following [50-54].

Random sampling, as well as LHS, choose samples directly from a continuous feature space, making it impossible to directly apply them to existing data sets. An oracle capable of online data labelling is required, covering not only the existing data points in a given data set, but continuously all possible samples in the range of that data set. Typically, a suitable simulation or real experiments can be used for this step. How- ever, simulations with adequate complexity and performance are rarely available out-of-the-box and using a real machine drastically limit the validation speed and excessively increases the efforts and costs [55-61].

To overcome this challenge, we fit a ground truth model (GTM) on the input real-world data set, which serves as an oracle later. A well trained GTM can approximate dependencies in the original data set and generalize it over continuous feature space within the observation limits used for training. This GTM not only reproduces labels for samples from the original data set, but also assigns a label to any sample in the complete continuous feature space of the original data, which enables the application of random sampling and LHS. This approach satisfies any sampling method that requires a continuous feature space [62-69].

The third method, namely QBC, cannot be used with GTM directly, because it requires a finite pool of unlabeled samples to iterate over in order to select next training samples (cf. Section II-B). Potentially any method that samples from a discrete finite pool, can use the input data set used for GTM training to do so. We do not follow this approach, because it has the following drawbacks. First of all, splitting the data set into train and test sets significantly reduces the number of available train samples [70-76].

Using the entire pool U for testing however distorts the result, because U shrinks over time and in the end only consists of those samples with the lowest information gain. Second, no assertions can be made about any input data set. It might (i) contain very unevenly distributed samples (bias), (ii) be too big and leaving the question of a meaningful subsampling open, or (iii) be complex and thus obscures the choice of a bootstrap train set. Third, while conducting experiments on a real physical system, often there is no to little prior data available. Knowledge how to initialize learning for AL methods in the most efficient way is a crucial part of AL use in real-world applications. Therefore, a robust validation schema for active learning should offer a high level of flexibility and control over the design of training and test data sets. Our validation approach avoids above-mentioned problems and allows for a flexible data collection [77-81].

B. Incremental Evaluation

Figure 5 outlines the comparison of data generation strategies. We divide it in four steps as explained below:

1) GTM fitting: This step consists of training an ML model on a chosen input data set. This step requires certain data- science background and includes data preprocessing, hyperopia remoter optimization of the ML model, validation of prediction accuracy and storing the trained GTM for the subsequent use as an oracle [1-4].

2) Data generation: Every iteration step from m = m + 1 is crucial, because choosing the sample with the highest information gain promises a better learning curve. Figure 6 illustrates how the different sampling methods extend their train set by selecting samples to be labeled. The input is a set of size m samples, which represents the so far known samples including their labels. The random method generates one random sample within predefined feature space X , statistically independent from any prior selections. This results in a set of m + 1 samples, which contains the prior m samples altogether with the newly selected one [5-9].

LHS generates from the feature space X a set of a specified, which cannot be changed or extended later. Therefore, LHS in our experiments discards the given input of m samples and creates a fresh one of size m + 1 independently from the input. For real-world applications, this means that the iterative extension of an existing data set designed with LHS is not possible and the number of experiments to conduct must be known in advanced. Despite this fact, there are application cases in which the number of experiments are predefined by time or cost constraints, making it relevant for our study. In this case, random sampling, LHS and AL are considered as equally-viable options in the process since only the final performance is relevant. The choice of a LHS over other extendable data generation strategies is indeed justified if it delivers good performance for the predefined amount of experiments [10-16].

The third method, namely AL, takes the input of m samples and selects the next sample from unlabeled set U for which its committee members disagree the most. We iterate over all still available unlabeled data $u \in U$ and let each committee member $\theta \in C$ make a prediction for this u. The function $s(\theta_1(u), \theta_2(u), ...)$ calculates the standard deviation of these predictions. That sample u with the highest standard deviation represents the largest disagreement of committee members and is selected. The result is a set of m+1 samples, which contains the prior m samples with one additional sample based on maximum disagreement [17-26].

3) Train and validate ML models: One test set of appropriate size is created via LHS in order to cover the whole input feature space and reliably evaluate the performance of predictive models trained during the evaluation. The prediction models are trained on train data sets as sampled by random sampling, LHS and AL and labelled by the GTM. The influence of every increment in train set size is validated on the test set, which is the same for all compared sampling methods. This allows a direct comparison of random sampling, LHS and AL in terms of sample efficiency, stability and predictive accuracy of trained models [27-33].

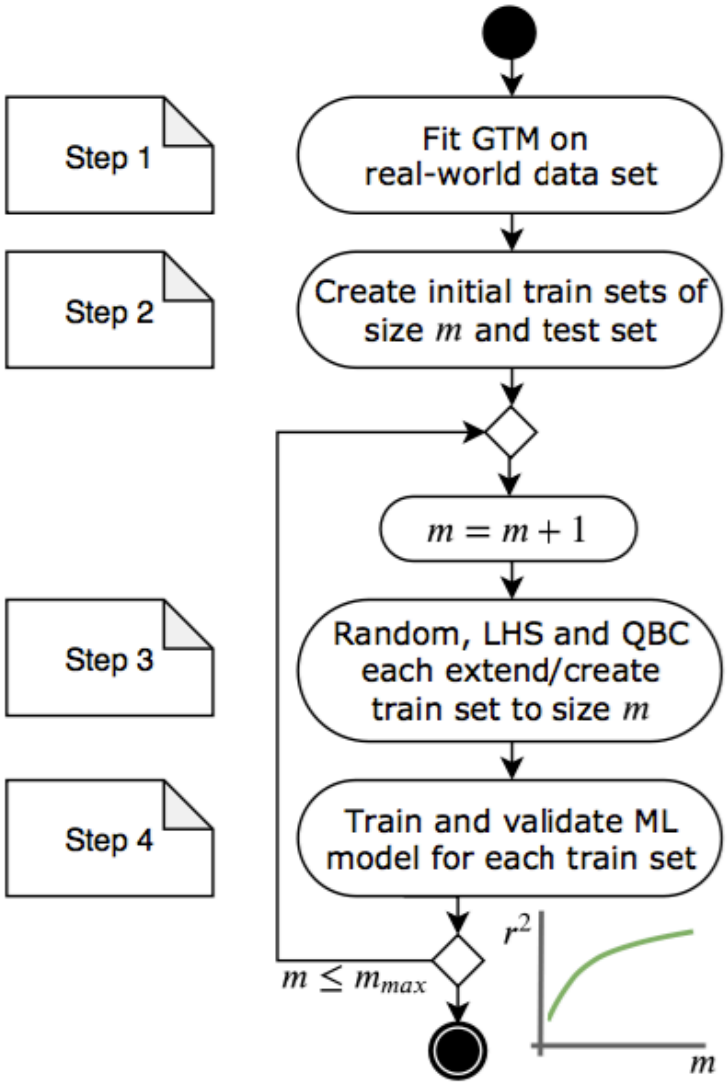


Fig. 5: We fit a GTM on a data set and use it to label samples requested by QBC, random and LHS. These grow with an increasing m and yield a validation plot. The entire procedure can be repeated multiple times (cross-validation) [67-73].

5. EXPERIMENTS

This section builds on the evaluation framework we developed in the prior section, executes the motivated steps and discusses their results.

A. Data sets for evaluation

Two main factors are used to select validation UCs: from one side the evaluation framework requires a labeled data set for GTM fitting. On the other hand, this paper focuses various sampling techniques for regression tasks, while approximating complex systems and processes in the field of mechanical engineering. The selected UCs base on publicly available data sets about complex mechanical systems.

1) Pumadyn family of data sets: data sets are generated by simulating the dynamic of a Puma 560 robot arm. Used simulation describes a robotic manipulator with a high degree of confidence. Pumadyn data sets have various level of complexity from linear to non-linear interaction with various levels of noise

in the measured signals. The data set allows to learn the dynamic model of the robotic arm and predict the angular acceleration of one of the robots arm links. Two data sets from the given family are used in this paper: *Pumadn-8nm* and *Pumadn-32nm*. Pumadyn-8nm includes 8 input features (angle positions of three links, angel velocities of three links, torque at first two joints) and one target variable (angle acceleration of joint 3). It consists of 8192 separate observations. Pumadyn-32nm offers 32 input features (angle positions of links 1 to 6, angel velocities of links 1 to 6, torques at joints 1 to 5, change in mass of links 1 to 5, change in length of links 1 to 5, change on viscous friction of links 1 to 5) with the same number of observations and angle acceleration of joint 6 as a target variable. Both data sets are highly non- linear and include a medium level of noise.

2) Electric Motor Temperature data set: it offers 36475 separate sensor measurements from a permanent magnet syn- chronous motor generated on a test bench. The purpose of the collected data is to use ambient temperature, coolant temperature, motors voltage, current, speed and torque to predict the surface temperature of the permanent magnet.

B. Machine Learning Models

In this paper, we work with two popular ML gradient boost- ing based algorithms: Extreme Gradient Boosting (XGBoost) and LightGBM (LGBM). Gradient boosting is a powerful ML technique reaching state-of-the-art results across a variety of practical task while been sample-efficient and relatively easy to implement. Both XGBoost and LGBM are machine tree-based learning algorithms that belong to the class of gradient boosting. The gradient boosting method trains several combined learning models with the data set to compensate for the weaknesses of each individual learning model. These cannot usually achieve higher accuracy on their own, which is why they are called "weak" learning models. Each new learning model focuses on previously mispredicted target values. The final prediction is based on the predictions of several "weak" learning models. This can significantly increase accuracy. Additionally, we look at how bagging of several predictive models improves the overall performance for different sampling strategies. Bagging can improve the performance by running many times an existing regression algorithm on a set of re-sampled data and averaging the predictions. Bagging is a good method for boosting the performance of decision-tree based algorithms from one side and can offer certain advantages while working with QBC sampling methods making it a good addition to our experimental setup.

C. Training and Validation of GTMs

XGBoost models [46] are used for GTM because of the achieved high prediction accuracy across all three data sets. Model tuning for every data set is conducted with the help of TPOT library [47] which utilizes evolutionary algorithms to find near-optimal model's settings within a user-defined range. Every model is validated with the ten-fold cross-validation (CV). During validation, the data set is shuffled and split into ten equal parts. Every part is used one time for validation of prediction accuracy against the real labels with the rest nine parts used for training. In total ten iterations referred to as CV folds are conducted. Prediction accuracy is evaluated using the explained variance score for every CV fold. Averaged explained variance score over all ten CV folds is taken as a final performance metric of a trained GTM. Tuned GTM trained on Pumadyn-8nm data set achieves explained variance score of 95.6%. Its prediction accuracy is visualized in Figure 7a. The x-axis represents real values and y-axis contains predicted values for the same input parameters obtained by CV. Ideal prediction lies on a line angled 45 degrees towards both axes since it means an exact match of predicted and real values. Prediction accuracy of tuned GTM trained on Pumadyn-32nm data set is visualized in Figure 7b. It achieves explained variance score of 91.7%. Tuned GTM trained on Electric Motor Temperature data set demonstrates an explained variance score of 96.4% and is visualized in Figure 7c. Visualized prediction performances and high explained variance scores prove that dependencies in initial data sets are fairly well approximated with tuned GTMs. All three GTM Models are trained on full-sized data sets with many thousand observations. Next step is to investigate what data sampling strategies allow to learn dependencies in the data with ML in the most sample-efficient way.

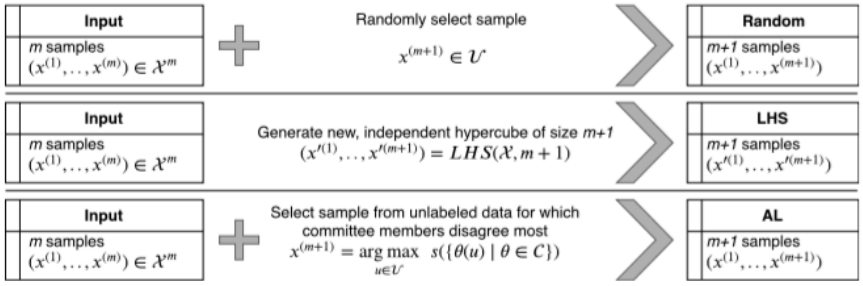


Fig. 6: Given m samples, the sampling strategies create $m+1$ samples differently. Random extends the given set by one random sample. LHS creates, independent from the given samples, a fresh set of $m+1$. QBC lets all committee members predict targets for all still unlabeled samples $\mathcal{U}$ and selects that with highest disagreement to extend the given m samples.

D. Comparison of Data Generation Strategies

We define important criteria for the comparison of data generation strategies as:

- *Prediction accuracy* of ML models trained on the generated data – all evaluation cases are regression tasks. We choose explained variance (R2) score as a performance metric since it intuitively explains how well a given ML model describes underlying dependencies in the data (100% – absolute accuracy, 0% – no dependencies learned)

- *Sample efficiency* – the rate of prediction accuracy in- crease with increasing train set size

Stability of results – is repeatability of observed prediction accuracy for the given number of samples over many iterations. Stability of results is reversed to the standard deviation of observed prediction accuracy for the given train size derived over many iterations. Random sampling, LHS and QBC techniques are compared on the three chosen UCs. An initial training data set L is generated with two data generation strategies: random sampling picks m random points within the valid feature space, LHS creates and experimental design of the same size. Those points are labeled by a GTM. Identical LGBM models are trained on both initial data sets. Performance of trained models is evaluated on a separate testing data set T of the size t. It incorporates LHS design and covers the same predefined feature space range. The same testing data set is used across all validation runs of considered UC in order to ensure precise and consistent evaluation of the prediction performance for all experiments. QBC sampling approach requires not only an initial train set but an unlabeled pool of data U of the size l, where new observations can be selected from. Both initial QBC train set L and pool set U are generated by means of LHS design. A committee consisting of three LGBM models is trained on bootstrapped parts of the training data. Subsequently, initial data sets are extended with additional k data points and the validation cycle is repeated again. Table II describes the validation setup for every validation UC.

TABLE II: Configuration of Training Setups

Validation UC	Initial train set size m	Pool set size l	Step size k	Test set size
Pumadyn-32nm	100	400	3	2000
Pumadyn-8nm	50	400	3	2000
ETM UC	50	400	3	2000

The main difference between experiments on different sampling techniques is the way new points are chosen. For random sampling corresponding training data set is repeatedly augmented with three randomly generated points within the defined feature space. It is important to note, that LHS excludes the possibility to subsequently refine an existing data set. For every required data size, a completely new train set is generated. It provides a solid baseline towards what prediction performance can be achieved with more sophisticated sampling methods, provided the training size is chosen and data is generated in one iteration only. In the case of QBC, a trained committee decides at each step which next observation points from the pool data set to label with GTM and include to the training data set. Apart from the training a LGBM model, we use a bagging predictive model consisting of three LGBM

models. A bagging predictive model can work particularly well while combined with a QBC sampling strategy based on a committee with equal compositions [45]. The experiment results are averaged over 30 independent runs for every setup (GTMs for three predictive tasks, three sampling techniques and two predictive models) including iterations from initial train set size to the final training size.

(a) Predicted vs. Real angle acceleration of joint 3, Pumadyn-8nm data set.

(b) Predicted vs. Real angle acceleration of joint 6, Pumadyn-32nm data set.

(c) Predicted vs. Real surface temperature of permanent magnet, Electric Motor Temperature data set.

Fig. 7: CV Results for GTMs.

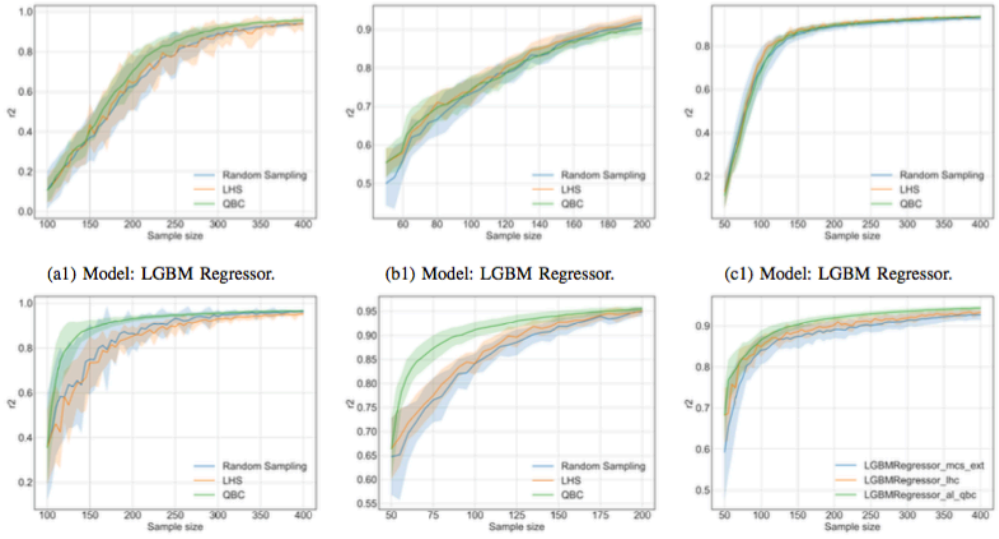
6. RESULTS

The evaluation results for all three validation UCs and two ML models are presented in Figure 8. These visualizations allow to compare various sampling methods over three defined comparison criteria:

- Points on represented learning curves depicted as solid lines stand for *prediction accuracy*.
- *Sample efficiency* is steepness of a given learning curve.

Stability of results can be evaluated through the standard deviation of observed accuracy values for the given number of training samples represented as the filled area around the given learning curve. For the Pumadyn-32nm validation UC (Figure 8a1 and Figure 8a2), random sampling has the highest deviation range between experimental runs for both single LGBM model and bagging model. QBC sampling approach performs equal to slightly better compared to both random sampling and LHS if a single LGBM model is trained. At the same time, the bagging model together with QBC noticeably surpasses all other data sampling approaches. It demonstrates a steeper improvement of prediction accuracy with growing train set size, has low deviations of prediction performance between validation runs and has a higher final explained variance score. As a next step, the efficiency of three data sampling meth- ods is investigated on a Pumadyn-8nm UC.

Looking at the Figure 8b1 and Figure 8b2 almost similar to the previous UC learning patterns can be seen. With a single LGBM model both QBC and LHS have an edge over random sampling in terms of result's stability and prediction performance at small training sample sizes. Bigger QBC-sampled training data, however, leads to a small prediction performance decrease comparing to other sampling methods. Bagging model combined with QBC sampling strategy, on the contrary, outperforms all other methods with a significant margin. The experiment on the Electric Motor Temperature data set leads to similar results. Related learning curves are visualized in Figure 8c1 and Figure 8c2. While with a single LGBM model QBC sampling can offer more consistent prediction performance over random sampling only, bagging model achieves faster and more stable learning with QBC.



(a1) Model: LGBM Regressor. (b1) Model: LGBM Regressor. (c1) Model: LGBM Regressor.

(a2) Model: Bagging. (b2) Model: Bagging. (c2) Model: Bagging.

(a) Validation UC: Pumadyn-32nm. (b) Validation UC: Pumadyn-8nm (c) Validation UC: EMT.

Fig. 8: Performance of random, LHS and AL Samplings over Three Validation UC

7. CONCLUSION

We considered the problem of AL readiness for real- world applications on an example of QBC sampling while designing data acquisition experiments in the field of mechanical engineering and manufacturing. We compared how QBC methods can help data-driven approximations of complex mechanical systems in comparison not only with the standard AL baseline - random sampling but as well with an established DOE method, namely LHS. This is particularly relevant for industrial applications since DOE is an industry-wide standard for planning and conducting data-collection experiments. This paper offers practical help for choosing data sampling strategies for particular experiments. We mentioned that using existing data sets for validation of new AL algorithms can lead to misleading results, because of potential bias introduced during the original data collection. Moreover, it does not allow conducting direct comparison to other sampling techniques such as DOE, which mostly place data points in the observation space by itself. We proposed a low-effort and sampling agnostic validation approach that is free of above- mentioned biases by introducing GTMs, which are trained on validation data sets. Three validation UCs were designed to investigate efficiency and result's stability of random, LHS and QBC sampling methods. We showed that, with a high number of experiments, the final results for all three methods are comparable. It is justified to use simple random sampling if the experimental costs are very low. When the number of available experiments is limited, LHS or AL prove to be a better choice. LHS delivers good results in terms of prediction accuracy and stability for all considered prediction models and sample sizes. However, LHS requires defining the sample count a- priori, therefore the number of experiments to execute must be chosen at an early planning stage. QBC offers an incremental extension of the train set and proved to be the best choice if certain conditions are met. For single prediction models, QBC needs tuning and trial iterations in order to achieve good results, therefore making it not suitable for applications with high costs for data acquisition. If the ML models intended for use on the collected data are compatible with ensembling techniques like bagging, e.g. LGBM, QBC is the most efficient and stable data sampling method with no tuning required.

REFRENCES

[1] Vartak, Manasi, et al. "ModelDB: a system for machine learning model management." Proceedings of the Workshop on Human-In-the-Loop Data Analytics. 2016.

[2] Schelter, Sebastian, et al. "On challenges in machine learning model management." (2015).

[3] Abdelaziz, Ahmed, et al. "A machine learning model for improving healthcare services on cloud computing environment." Measurement 119 (2018): 117-128.
[4] Wang, Chengcheng, et al. "Machine learning in additive manufacturing: State-of-the-art and perspectives." Additive Manufacturing 36 (2020): 101538.
[5] Zhu, Zuowei, et al. "Machine learning in tolerancing for additive manufacturing." CIRP Annals 67.1 (2018): 157-160.
[6] Shafaie, Majid, Maziar Khademi, Mohsen Sarparast, and Hongyan Zhang. "Modified GTN parameters calibration in additive manufacturing of Ti-6Al-4 V alloy: a hybrid ANN-PSO approach." The International Journal of Advanced Manufacturing Technology 123, no. 11 (2022): 4385-4398.
[7] Meng, Lingbin, et al. "Machine learning in additive manufacturing: a review." Jom 72.6 (2020): 2363-2377.
[8] Sarparast, Mohsen, Majid Ghoreishi, Tohid Jahangirpoor, and Vahid Tahmasbi. "Experimental and finite element investigation of high-speed bone drilling: evaluation of force and temperature." Journal of the Brazilian Society of Mechanical Sciences and Engineering 42, no. 6 (2020): 1-9.
[9] Li, Zhixiong, et al. "Prediction of surface roughness in extrusion-based additive manufacturing with machine learning." Robotics and Computer-Integrated Manufacturing 57 (2019): 488-495.
[10] Jin, Zeqing, et al. "Machine learning for advanced additive manufacturing." Matter 3.5 (2020): 1541-1556.
[11] Jiang, Jingchao, et al. "Machine learning integrated design for additive manufacturing." Journal of Intelligent Manufacturing (2020): 1-14.
[12] Sarparast, Mohsen, Majid Ghoreishi, Tohid Jahangirpoor, and Vahid Tahmasbi. "Modelling and optimisation of temperature and force behaviour in high-speed bone drilling." Biotechnology & Biotechnological Equipment 33, no. 1 (2019): 1616-1625.
[13] Baumann, Felix W., et al. "Trends of machine learning in additive manufacturing." International Journal of Rapid Manufacturing 7.4 (2018): 310-336.
[14] Gu, Grace X., et al. "Bioinspired hierarchical composite design using machine learning: simulation, additive manufacturing, and experiment." Materials Horizons 5.5 (2018): 939-945.S
[15] Bahrami, Javad, Mohammad Ebrahimabadi, Sofiane Takarabt, Jean-luc Danger, Sylvain Guilley, and Naghmeh Karimi. "On the Practicality of Relying on Simulations in Different Abstraction Levels for Pre-Silicon Side-Channel Analysis."
[16] Qi, Xinbo, et al. "Applying neural-network-based machine learning to additive manufacturing: current applications, challenges, and future perspectives." Engineering 5.4 (2019): 721-729.
[17] Johnson, N. S., et al. "Invited review: Machine learning for materials developments in metals additive manufacturing." Additive Manufacturing 36 (2020): 101641.
[18] Sadi, Mehdi, Yi He, Yanjing Li, Mahabubul Alam, Satwik Kundu, Swaroop Ghosh, Javad Bahrami, and Naghmeh Karimi. "Special Session: On the Reliability of Conventional and Quantum Neural Network Hardware." In 2022 IEEE 40th VLSI Test Symposium (VTS), pp. 1-12. IEEE, 2022.
[19] Qin, Jian, et al. "Research and application of machine learning for additive manufacturing." Additive Manufacturing (2022): 102691.
[20] Caggiano, Alessandra, et al. "Machine learning-based image processing for on-line defect recognition in additive manufacturing." CIRP annals 68.1 (2019): 451-454.
[21] Bahrami, Javad, Mohammad Ebrahimabadi, Jean-Luc Danger, Sylvain Guilley, and Naghmeh Karimi. "Leakage power analysis in different S-box masking protection schemes." In 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1263-1268. IEEE, 2022.
[22] Yao, Xiling, Seung Ki Moon, and Guijun Bi. "A hybrid machine learning approach for additive manufacturing design feature recommendation." Rapid Prototyping Journal (2017).
[23] Bahrami, Javad, Viet B. Dang, Abubakr Abdulgadir, Khaled N. Khasawneh, Jens-Peter Kaps, and Kris Gaj. "Lightweight implementation of the lowmc block cipher protected against side-channel attacks." In Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security, pp. 45-56. 2020.
[24] Sing, S. L., et al. "Perspectives of using machine learning in laser powder bed fusion for metal additive manufacturing." Virtual and Physical Prototyping 16.3 (2021): 372-386.
[25] Ahmadinejad, Farzad, Javad Bahrami, Mohammd Bagher Menhaj, and Saeed Shiry Ghidary. "Autonomous Flight of Quadcopters in the Presence of Ground Effect." In 2018 4th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), pp. 217-223. IEEE, 2018.
[26] Ko, Hyunwoong, et al. "Machine learning and knowledge graph based design rule construction for additive manufacturing." Additive Manufacturing 37 (2021): 101620.
[27] Hadiana, Hengameh, Amir Mohammad Golmohammadib, Hasan Hosseini Nasabc, and Negar Jahanbakhsh Javidd. "Time Parameter Estimation Using Statistical Distribution of Weibull to Improve Reliability." (2017).
[28] Jiang, Jingchao, et al. "Achieving better connections between deposited lines in additive manufacturing via machine learning." Math. Biosci. Eng 17.4 (2020): 3382-3394.
[29] Zavareh, Bozorgasl, Hossein Foroozan, Meysam Gheisarnejad, and Mohammad-Hassan Khooban. "New trends on digital twin-based blockchain technology in zero-emission ship applications." Naval Engineers Journal 133, no. 3 (2021): 115-135.
[30] Li, Rui, Mingzhou Jin, and Vincent C. Paquit. "Geometrical defect detection for additive manufacturing with machine learning models." Materials & Design 206 (2021): 109726.
[31] Baturynska, Ivanna, 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 (2018): 227-232.
[32] Bozorgasl, Zavareh, and Mohammad J. Dehghani. "2-D DOA estimation in wireless location system via sparse representation." In 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 86-89. IEEE, 2014.
[33] Gobert, Christian, et al. "Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging." Additive Manufacturing 21 (2018): 517-528.
[34] Sullivan, Emily. "Understanding from machine learning models." The British Journal for the Philosophy of Science (2022).
[35] Paul, Arindam, et al. "A real-time iterative machine learning approach for temperature profile prediction in additive manufacturing processes." 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE, 2019.
[36] Amini, Mahyar, and Aryati Bakri. "Cloud computing adoption by SMEs in the Malaysia: A multi-perspective framework based on DOI theory and TOE framework." Journal of Information Technology & Information Systems Research (JITISR) 9.2 (2015): 121-135.
[37] Mahmoud, Dalia, et al. "Applications of Machine Learning in Process Monitoring and Controls of L-PBF Additive Manufacturing: A Review." *Applied Sciences* 11.24 (2021): 11910.
[38] Amini, Mahyar. "The factors that influence on adoption of cloud computing for small and medium enterprises." (2014).
[39] Khanzadeh, Mojtaba, et al. "Quantifying geometric accuracy with unsupervised machine learning: Using self-organizing map on fused filament fabrication additive manufacturing parts." *Journal of Manufacturing Science and Engineering* 140.3 (2018).
[40] Amini, Mahyar, et al. "MAHAMGOSTAR.COM AS A CASE STUDY FOR ADOPTION OF LARAVEL FRAMEWORK AS THE BEST PROGRAMMING TOOLS FOR PHP BASED WEB DEVELOPMENT FOR SMALL AND MEDIUM ENTERPRISES." Journal of Innovation & Knowledge, ISSN (2021): 100-110.
[41] Liu, Rui, Sen Liu, and Xiaoli Zhang. "A physics-informed machine learning model for porosity analysis in laser powder bed fusion additive manufacturing." *The International Journal of Advanced Manufacturing Technology* 113.7 (2021): 1943-1958.
[42] Amini, Mahyar, et al. "Development of an instrument for assessing the impact of environmental context on adoption of cloud computing for small and medium enterprises." Australian Journal of Basic and Applied Sciences (AJBAS) 8.10 (2014): 129-135.
[43] Nojeem, Lolade, et al. "Machine Learning assisted response circles for additive manufacturing concerning metal dust bed fusion." International Journal of Management System and Applied Science 17.11 (2023): 491-501.
[44] Golmohammadi, Amir-Mohammad, Negar Jahanbakhsh Javid, Lily Poursoltan, and Hamid Esmaeeli. "Modeling and analyzing one vendor-multiple retailers VMI SC using Stackelberg game theory." Industrial Engineering and Management Systems 15, no. 4 (2016): 385-395.
[45] Motalo, Kubura, et al. "evaluating artificial intelligence effects on additive manufacturing by machine learning procedure." Journal of Basis Applied Science and Management System 13.6 (2023): 3196-3205.
[46] Fu, Yanzhou, et al. "Machine learning algorithms for defect detection in metal laser-based additive manufacturing: a review." *Journal of Manufacturing Processes* 75 (2022): 693-710.
[47] Amini, Mahyar, et al. "The role of top manager behaviours on adoption of cloud computing for small and medium enterprises." Australian Journal of Basic and Applied Sciences (AJBAS) 8.1 (2014): 490-498.
[48] Stathatos, Emmanuel, and George-Christopher Vosniakos. "Real-time simulation for long paths in laser-based additive manufacturing: a machine learning approach." *The International Journal of Advanced Manufacturing Technology* 104.5 (2019): 1967-1984.
[49] Amini, Mahyar, and Nazli Sadat Safavi. "Critical success factors for ERP implementation." International Journal of Information Technology & Information Systems 5.15 (2013): 1-23.
[50] Petrich, Jan, et al. "Multi-modal sensor fusion with machine learning for data-driven process monitoring for additive manufacturing." *Additive Manufacturing* 48 (2021): 102364.
[51] Amini, Mahyar, et al. "Agricultural development in IRAN base on cloud computing theory." International Journal of Engineering Research & Technology (IJERT) 2.6 (2013): 796-801.
[52] Snow, Zackary, et al. "Toward in-situ flaw detection in laser powder bed fusion additive manufacturing through layerwise imagery and machine learning." *Journal of Manufacturing Systems* 59 (2021): 12-26.
[53] Amini, Mahyar, et al. "Types of cloud computing (public and private) that transform the organization more effectively." International Journal of Engineering Research & Technology (IJERT) 2.5 (2013): 1263-1269.
[54] Akhil, V., et al. "Image data-based surface texture characterization and prediction using machine learning approaches for additive manufacturing." *Journal of Computing and Information Science in Engineering* 20.2 (2020): 021010.
[55] Amini, Mahyar, and Nazli Sadat Safavi. "Cloud Computing Transform the Way of IT Delivers Services to the Organizations." International Journal of Innovation & Management Science Research 1.61 (2013): 1-5.
[56] Douard, Amélina, et al. "An example of machine learning applied in additive manufacturing." *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2018.

[57] Amini, Mahyar, and Nazli Sadat Safavi. "A Dynamic SLA Aware Heuristic Solution For IaaS Cloud Placement Problem Without Migration." International Journal of Computer Science and Information Technologies 6.11 (2014): 25-30.
[58] Ding, Donghong, et al. "The first step towards intelligent wire arc additive manufacturing: An automatic bead modelling system using machine learning through industrial information integration." *Journal of Industrial Information Integration* 23 (2021): 100218.
[59] Amini, Mahyar, and Nazli Sadat Safavi. "A Dynamic SLA Aware Solution For IaaS Cloud Placement Problem Using Simulated Annealing." International Journal of Computer Science and Information Technologies 6.11 (2014): 52-57.
[60] Zhang, Wentai, et al. "Machine learning enabled powder spreading process map for metal additive manufacturing (AM)." *2017 International Solid Freeform Fabrication Symposium*. University of Texas at Austin, 2017.
[61] Sharifani, Koosha and Amini, Mahyar and Akbari, Yaser and Aghajanzadeh Godarzi, Javad. "Operating Machine Learning across Natural Language Processing Techniques for Improvement of Fabricated News Model." International Journal of Science and Information System Research 12.9 (2022): 20-44.
[62] Du, Y., T. Mukherjee, and T. DebRoy. "Physics-informed machine learning and mechanistic modeling of additive manufacturing to reduce defects." *Applied Materials Today* 24 (2021): 101123.
[63] Sadat Safavi, Nazli, et al. "An effective model for evaluating organizational risk and cost in ERP implementation by SME." IOSR Journal of Business and Management (IOSR-JBM) 10.6 (2013): 70-75.
[64] Chen, Lequn, et al. "Rapid surface defect identification for additive manufacturing with in-situ point cloud processing and machine learning." *Virtual and Physical Prototyping* 16.1 (2021): 50-67.
[65] Sadat Safavi, Nazli, Nor Hidayati Zakaria, and Mahyar Amini. "The risk analysis of system selection and business process re-engineering towards the success of enterprise resource planning project for small and medium enterprise." World Applied Sciences Journal (WASJ) 31.9 (2014): 1669-1676.
[66] Yaseer, Ahmed, and Heping Chen. "Machine learning based layer roughness modeling in robotic additive manufacturing." *Journal of Manufacturing Processes* 70 (2021): 543-552.
[67] Sadat Safavi, Nazli, Mahyar Amini, and Seyyed AmirAli Javadinia. "The determinant of adoption of enterprise resource planning for small and medium enterprises in Iran." International Journal of Advanced Research in IT and Engineering (IJARIE) 3.1 (2014): 1-8.
[68] Chen, Lequn, et al. "Rapid surface defect identification for additive manufacturing with in-situ point cloud processing and machine learning." Virtual and Physical Prototyping 16.1 (2021): 50-67.
[69] Safavi, Nazli Sadat, et al. "An effective model for evaluating organizational risk and cost in ERP implementation by SME." IOSR Journal of Business and Management (IOSR-JBM) 10.6 (2013): 61-66.
[70] Guo, Shenghan, et al. "Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm." Journal of Manufacturing Systems 62 (2022): 145-163.
[71] Khoshraftar, Alireza, et al. "Improving The CRM System In Healthcare Organization." International Journal of Computer Engineering & Sciences (IJCES) 1.2 (2011): 28-35.
[72] Ren, K., et al. "Thermal field prediction for laser scanning paths in laser aided additive manufacturing by physics-based machine learning." Computer Methods in Applied Mechanics and Engineering 362 (2020): 112734.
[73] Abdollahzadegan, A., Che Hussin, A. R., Moshfegh Gohary, M., & Amini, M. (2013). The organizational critical success factors for adopting cloud computing in SMEs. Journal of Information Systems Research and Innovation (JISRI), 4(1), 67-74.
[74] Zhu, Qiming, Zeliang Liu, and Jinhui Yan. "Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks." Computational Mechanics 67.2 (2021): 619-635.
[75] Amini, Mahyar and Ali Rahmani. "Agricultural databases evaluation with machine learning procedure." Australian Journal of Engineering and Applied Science 8.6 (2023): 39-50.
[76] Newiduom, Ladson, et al. "Machine learning model improvement based on additive manufacturing process for melt pool warmth predicting." International Journal of Applied Science and Engineering Research 15.6 (2023): 240-250.
[77] Jackson, Keypi, et al. "Increasing substantial productivity in additive manufacturing procedures by expending machine learning method." International Journal of Mechanical Engineering and Science Study 10.8 (2023): 363-372.
[78] Opuiyo, Atora, et al. "Evaluating additive manufacturing procedure which was using artificial intelligence." International Journal of Science and Information System 13.5 (2023): 107-116.
[79] Amini, Mahyar, and Ali Rahmani. "Machine learning process evaluating damage classification of composites." Journal 9.12 (2023): 240-250.
[80] Yun, Chidi, et al. "Machine learning attitude towards temperature report forecast in additive manufacturing procedures." International Journal of Basis Applied Science and Study 9.12 (2023): 287-297.
[81] Shun, Miki, et al. "Incorporated design aimed at additive manufacturing through machine learning." International Journal of Information System and Innovation Adoption 16.10 (2023): 823-831.