NORTHWESTERN UNIVERSITY

Machine Learning and Data-driven Optimization for Applications in Scientific Discovery

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Engineering

By

Arindam Paul

EVANSTON, ILLINOIS

December 2019

ABSTRACT

Data Mining is a multidisciplinary science combining concepts from machine learning, statistics, and database systems to address problems using universal principles based on mathematical hypotheses. It involves understanding and interpreting data from any source and extracting useful information. A data mining approach can convert data of any form, and thus, it has been successful in solving problems across diverse applications including social media, finance, medical and scientific applications. This thesis discusses development of data-driven solutions for knowledge discovery in scientific applications.

Understanding the processing-structure-property-performance relationships are essential for scientific discovery and prediction. A forward problem is based on an understanding relationship between processing method, and it's subsequent structure and the effect on the resultant property and performance. An inverse problem is an approach of optimizing or reverse engineering the desirable structure and the processing involved with a desired property and performance. For applications in scientific computing, this translates to property prediction and knowledge discovery using data-driven methods. This dissertation proposal attempts to advance data mining and knowledge discovery methodologies to solve some of the challenging sub-problems in the domain of both forward and inverse problems pertaining to scientific applications.

Acknowledgements

I would first like to thank Prof. Alok Choudhary and Prof. Ankit Agrawal for advising me during the program. I would also like to thank my committee member Prof. Wei-keng Liao for his continuous support and suggestions. Next, I would take the opportunity to thank the funding agencies whose grants supported my thesis. This thesis was supported by the NIST awards 70NANB14H012 and 70NANB19H005 from U.S. Department of Commerce, National Institute of Standards and Technology as part of the Center for Hierarchical Materials Design (CHiMaD); AFOSR MURI award FA9550-12-1-0458, NSF award CCF-1409601; DOE awards DE-SC0007456, DE-SC0014330 and DE-SC0019358; and Northwestern Data Science Initiative.

It has been a privilege to work alongside outstanding researchers at Northwestern University, and I am grateful for this opportunity. I would also like to thank Prof. Bruce Lindvall and Prof. Allen Taflove for their help in their administrative capacities when I had to change research labs during my PhD. I would also like to thank Prof. Jeremy Birnholtz and Prof. Noshir Contractor for an opportunity to collaborate in the domain of social media and analytics. I would also like to thank my first academic advisor at Northwestern University, Prof. Aleksandar Kuzmanovic for giving me the opportunity to start my PhD at Northwestern University. I offer my gratitude to Prof. Doug Downey for his research advice in machine learning and natural language processing. I would also like

to thank Dr. Alona Furmanchuk who helped me as a mentor in cheminformatics when I was starting to work in this area.

I would also like to thank my therapist Dr. M (full name not divulged for anonymity) as this dissertation would never have been possible without her help. During my PhD, I have gone through very dark days suffering in a cocktail of anxiety, depression and loneliness. She was instrumental in providing professional guidance and helped me overcome all mental and emotional difficulties. I would like to thank Northwestern Counseling and Psychological Services (CAPS) for the counseling, therapy and meditation services they provide for graduate students. In particular, I would like to thank Dr. Wei-Jen Huang at CAPS for introducing me to self-care and his profound advice and kind words.

I would want to thank my lab-mates for making the graduate school years enjoyable and memorable. In particular, I would like to mention Dipendra Jha and Reda Al-Bahrani who were not just lab-mates but very good friends. I would miss the long excited conversations as I move on to the next phase of my life. I would like to thank my room-mates who were all Northwestern students for their incredible support. In particular, I am indebted to Divya Jain and Bharath Pattabiraman who were a pillar of strength especially during the most tumultuous times of my PhD. Divya had been like a constant friend and Bharath like an older brother during my time as a PhD student. I would also like to thank some other friends at Northwestern University Mas-ud Hussain, Lisa Buchter and Emily Harburg for their constant support during my PhD. During my time at Northwestern, I had been a part of the Toastmasters chapter at Northwestern, the creative writing clubs at Meetup and Northwestern, the multi-cultural dialogue group and the tango club at different points of my PhD. I would like to thank my roommates

Siddhartha and Nadim for their "tough love" which made me stronger emotionally, and also for teaching me driving. I would also like to express gratitude to my friend, Mohit, whom I have known since middle school for being there when I needed him.

My amazing family has been a source of endless encouragement and motivation in this pursuit. I would like to thank my parents who always encouraged me to be a scientist from a very small age. At the age of five years, I could imagine doing a PhD one day and that was due to the vision provided by my parents. My sister Panchali and brother-in-law Jibat Sankar also provided guidance and reassurance during the course of my PhD. I wish to thank my loving fiance, Roumita, who has provided constant inspiration and has been my side during the final and most crucial steps of my PhD.

There are so many others both during my time at Northwestern University and before. One's dissertation is like a child for every doctoral student. It requires care and nurture and patience for years to raise a child. But, also, *it takes a village to raise a child* and therefore, it took motivation, advice, guidance from at least hundreds of people during my lifetime to make this happen. Every small tip, every tiny suggestion, every little word of encouragement have helped make this happen. During times of crisis, when I had considered leaving the PhD, all these have helped me stay steady on the course.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Problem Description and Research Objective

1.1. Data Mining and Scientific Applications

Traditionally, mechanical engineers, chemists and materials scientists relied on experimentally generated or simulation-based computational data to discover new materials and understand their characteristics. In the past decade, there has been a growing interest and shift towards data-driven scientifc discovery [4, 5]. It has stimulated researchers' interest in the application of advanced data-driven based machine learning techniques for accelerated discovery and design of materials and chemical compounds, supported by the Materials Genome Initiative (MGI) [6]. MGI intends to half the cycle time of development of new materials to implementation in real-world cases - from 20 yrs to 10 yrs. Data-driven techniques provide faster methods to know important properties of chemical compounds and to predict feasibility to experimentally synthesize in chemical laboratories, and thus promises to accelerate the research process of new materials development.

There have been many initiatives to assist scientific discovery using machine learning techniques [7–12]. Ward et al. [8] used random forests (RF) and clustering for discovering new photovoltaic materials and metallic glass alloys. Agrawal et al. [11] used linear, polynomial and support vector machine (SVM) regression for the prediction of fatigue strength of steels from their composition and processing parameters. Liu et al. [12] used decision trees and SVMs for reducing the search space using feature selection for microstructure

optimization. Xue et al. [9] used support vector regression (SVR) to infer the thermal hysteresis of NiTi-based shape memory alloys to accelerate the search for materials with low thermal hysteresis properties. Further, deep neural networks have gained significant attention and enjoyed success in applications in materials science [13–18].

1.2. Challenges

Data Mining for scientific discovery provides a unique set of challenges. The data is not only heterogeneous, multi-scale, multi-dimensional, and collected from multiple contexts, but also often, the datasets can be much smaller than comparable datasets in other domains. It is essential to design the problem into an appropriate form, process the data appropriately, and craft suitable models that fit both the nature of the problem and that of the data, and thereby, improving the accuracy and reliability of the models. Few of the broad challenges encountered in developing solutions for the two broad classes of problems in materials mining: forward (property prediction) and inverse (materials discovery) [4] are discussed.

1.2.1. Forward Problems

Processing-structure-property-performance (PSPP) [19] relationships form the backbone of discovery and prediction of materials. How a certain material is processed decides its subsequent structure, and that, in turn, impacts the property, and the performance of the device developed using that material. The molecular structure of a chemical compound determine the properties exhibited by that compound, and in turn, impact where and how

that compound can be used. The forward materials problem translates into a property prediction problem with various hierarchies of materials structure impacting the property.

1.2.1.1. Challenges in mining small high-dimensional datasets. There are two fundamental sources of materials data: experimental and computational data. However, one of the most significant challenges, in particular for experimental datasets, is the small size of datasets. Experiments are expensive with respect to both human and scientific resources. However, many of the feature sets have thousands of features. For instance, molecular fingerprints (Chapters 2, 3 and 4) are one type of representation used to describe organic molecules, and the dimensionality of the feature set can vary from 2000 to 4 million. The essence of supervised learning is based on creating a model that can learn the relationship between the parameters and the target label. However, when the dataset is much smaller compared to the feature set, learning the parameters become difficult due to the curse of dimensionality [20].

1.2.1.2. Challenges in data representation. One of the challenges with mining scientific datasets is that one source of data may not be sufficient for predicting the desired property. However, this is not directly comprehensible at the commencement of the mining process. This requires an iterative process of feature development to augment the model with additional information, and involves organizing and manipulating multiple forms of data and representing in a way that is suitable for learning (Chapters 2, 4 and 5).

1.2.1.3. Challenges in cost-effective modeling. Another challenge in developing many predictive modeling systems is keeping the training time low. Models based on algorithms such as deep neural networks are very powerful and flexible, and are able to

successfully develop accurate models for almost any kind of prediction problem. However, most of these models suffer are expensive in terms of training time. This becomes even more critical when machine learning is used for developing surrogate models for scientific simulations. In such cases, a model needs to be fast to train to be useful (Chapter 5).

1.2.2. Inverse Problems

While the forward PSPP problems deal with deducing the property given the processing method and structure, the inverse problem entails suggesting and discovering structures and processing techniques for a desired property or performance. This is usually more challenging as the relationship in the forward directions is often many to one, and therefore, there are no inverse functions. Multiple processing techniques can produce the same structure, and many distinct structures can lead to the same property.

1.2.2.1. Challenges in searching high-dimensional materials spaces. One of the challenging and essential inverse problems in materials science is aiming to obtain the complete set of all possible microstructures for optimizing a given property. Forward models can compute properties for a given microstructure, but discovering all possible microstructures satisfying a property is expensive. Exhaustive search based methods would be computationally expensive. Further, methods like pattern search or trust region search fail to converge as these methods depend on iteratively searching for points that converge. Therefore, it becomes imperative to develop methodologies that can provide optimal solutions without becoming prohibitively expensive based on computational resource demands (Chapters 6 and 7).

1.3. Dissertation Problem Statement

The dissertation problem statement is *"to develop data-driven optimization and machine learning systems for knowledge discovery in scientific applications"*. This dissertation focuses on the development of systems for prediction and optimization tasks for scientific applications in disciplines such as chemistry, materials science, aeronautics and manufacturing. These applications employ several techniques such as sequence prediction, multi-input single-output (MISO) modeling and iterative machine learning. In this dissertation, three lines of work are pursued. The first discusses dealing with feature manipulation, ensemble learning, multiple representation learning for property prediction of molecular compounds. The second discusses a real-time iterative machine learning based surrogate model to predict the temperature profile for an additive manufacturing process. The third discusses discovering numerous distinct optimal solutions by narrowing the search space of constrained design problems. These three applications employ several techniques including synthetic dataset generation, feature reduction, and selection, feature manipulation, ensemble learning as well as deep learning.

1.4. Thesis Organization

In this thesis, we explore how data-driven techniques can be used to solve a wide variety of scientific discovery problems. We present two types of work, one that resorts to predictive modeling, and the other that concerns with optimization. A total of six works are presented in this thesis. The first four works focus on developing machine learning based prediction models known as forward modeling. The last two works focus on development of data-driven optimization known as inverse modeling.

The first three works relates to development of machine learning models on organic molecules. The first work (Chapter 2) concerns development of a multi input single output deep neural architecture for predicting chemical properties across several benchmark datasets. This work utilizes both text and vector-based descriptions of molecules as inputs to the neural networks. The second work (Chapter 3) propounds a tree-based ensemble model using extremely randomized trees for prediction of solar cell band gaps from a very small dataset of 350 molecules. As the model is based on ensemble of trees, it allows for deeper analysis of which part of the molecule is responsible for the property. The third work (Chapter 4) utilizes pieces from the first and second work. A transfer learning approach is undertaken for prediction of organic solar cell properties. An initial model is trained on a large database of first principle based calculations of hypothetical molecules. The pretrained model is then trained on the small dataset of 350 molecules. It utilizes two different text based representations as inputs.

The fourth work (Chapter 5) develops a machine learning based surrogate model for additive manufacturing simulations. Additive Manufacturing (AM) [21, 22] techniques are becoming extremely popular due to the capability to produce elaborate designs as well as fast prototyping. Although AM may be used for rapid prototyping, commonly referred to as 3D printing, it is also used for developing finished product using techniques such as Direct Metal Deposition. It becomes imperative to design a tool for simulating the process. However, traditional finite element method (FEM)-based simulation based tools take a lot of computational resources and time. In this work, a predictive model is developed to form an essential component of a scientific framework for a ML-based real-time control system of additive manufacturing.

Finally, the fifth and sixth works develop a data-driven solution for constrained microstructure optimization problems. Traditional optimization algorithms such as linear programming are able to find only a single optimal solution. Algorithms such as genetic programming take a lot of time for converging. Black box algorithms such as pattern search are unable to converge because of the number of constraints in the problem space. In the fifth work (Chapter 6), a data-driven approach based on sampling statistics were employed to reduce the search space and converge on many optimal and near-optimal solutions. In the sixth work (Chapter 7), decision trees were employed to iteratively reduce the search space by creating candidate solutions, extracting sub-optimal solution and then repeating the process.

CHAPTER 2

Predicting Chemical Properties using Mixed Deep Neural Networks and Multiple Molecular Representations

2.1. Introduction

Traditionally, chemists and materials scientists have relied on experimentally generated or simulation-based computational data to discover new materials and understand their characteristics. The slow pace of development and deployment of new/improved materials has been considered as the main bottleneck in the innovation cycles of most emerging technologies [23]. Data-driven techniques provide faster methods to identify important properties of chemical compounds and to predict feasibility to synthesize in chemical laboratories and thus promise to accelerate the research process of new materials development. There have been many initiatives to computationally assist molecular and materials discovery using machine learning (ML) techniques [4, 7, 8, 11, 24–27]. Conventional machine learning approaches for predicting chemical properties have emphasized the importance of leveraging domain knowledge when designing model inputs. Current research has demonstrated that deep neural networks (DNNs) have generally outperformed traditional machine learning models. DNN models are capable of learning representations, which sets it apart from conventional ML algorithms used in chemistry. Representation learning is the process of transforming input data into a set of features that can be effectively exploited to identify patterns from the data. In the context of

chemistry, the analogous process would be to use deep learning (DL) to examine chemical structures and to construct features similar to engineered chemical features, with minimal assistance from an expert chemist. This approach that leverages representation learning of deep neural networks is a significant departure from the traditional research paradigm in chemistry.

In this work, we develop CheMixNet- a set of neural networks for predicting chemical properties by leveraging multiple molecular representations as inputs. We used simplified molecular-input line-entry system (SMILES) [28] notations as sequence inputs and molecular fingerprints as vector inputs. SMILES is a line notation of chemical structures which encodes the connection table and the stereochemistry of a molecule as a line of text. Our work improves upon the existing state-of-the-art approach of directly learning from vector representations such as molecular fingerprints or chemical text representations such as SMILES by harnessing the network structure of both forms of representations. CheMixNet is a variation of multi-input-single-output (MISO) [29] architectures that learn the chemical properties from a mix of intermediate features learned from two different input representations - a vector input in the form of molecular fingerprints and a sequence input in the form of SMILES strings. In our experiments, we used MACCS fingerprints - a first 2D representation of chemical structure using 167 features. Although MACCS usually perform worse than other molecular fingerprints, we chose MACCS because of its simplicity and ease of interpretation. We perform significant experimentation to determine the best neural network structure for the CheMixNet architectures.

We evaluated the effectiveness of our mixed approach for building DNN architectures by training CheMixNet on six different datasets- a large dataset composed of 2.3 million

samples from the Harvard Clean Energy Project (CEP) database and five other relatively smaller datasets from the MoleculeNet [30, 31] benchmark. Compared to other DL models, CheMixNet architecture outperforms fully connected MLP models trained on molecular fingerprints, recurrent neural networks (RNN) and 1-dimensional convolutional neural network (CNN) models trained on SMILES, as well as other models - convolutional molecular graphs (ConvGraph) [32] and Chemception [33]. For instance, we achieved a mean absolute percentage error (MAPE) of 0.24 % on the CEP dataset; this is significantly better than the MAPE of 0.43 % using CNN-RNN model. The CheMixNet architectures, as well as the benchmark models, are made accessible for the research community at https://github.com/paularindam/CheMixNet [34].

2.2. Background & Related Works

In this section, we present a description of the two molecular representations we use in this work - SMILES and molecular fingerprints, and discuss existing deep neural architectures for predictive modeling of chemical properties in the Quantitative structure-activity relationship (QSAR)/Quantitative structure-property relationship (QSPR) [35] modeling.

2.2.1. SMILES & Fingerprints

Line notations are linear representations of chemical structures which encode the connection table and the stereochemistry of a molecule as a line of text [36]. SMILES [28] is the most popular specification in the form of a line notation to describe the structure of chemical species using short ASCII strings encoding molecular structures and specific instances. One or more organic molecules attach to form long continuous chains known

as branches. SMILES has a grammar structure in which alphabets denote atoms, special characters such as = and ≡ bond denote the type of bonds, encapsulated numbers indicate rings, and parentheses represent side chains. In this work, we limit ourselves to character level representation and do not explicitly encode the grammar.

Molecular fingerprints are representations of chemical structures, successfully used in similarity search [37], clustering [38], classifications [39], drug discovery [40], and virtual screening [41], a standard and computationally efficient abstract representation where structural features are represented by either bits in a bit string or counts in a count vector. Fingerprints were motivated by the need to find materials that match target material properties. They follow the assumptions that the properties of the material is a direct function of its structure and that materials with similar structure are likely to have similar physical-chemical character. Different fingerprints represent different aspects of a molecule, and thus each type of fingerprint can have different suitability for mapping to particular physical property. Various machine learning (ML) algorithms have been used to predict the activity or property of chemicals using molecular descriptors and/or fingerprints as input features. In our experiments, we used MACCS fingerprints [42, 43] - a primitive 2D representation of chemical structure using 167 features. MACCS fingerprints were originally developed for the purpose of substructure screening. Unlike other hashed 1024 bit fingerprint representations such as Atom Pair and Topological Torsion that are difficult to comprehend, MACCS fingerprints represent the counts of the presence or absence of chemical fragments, and are easily comprehensible with each key having its own definition (e.g. key 99 indicates if there is a C=C bond, key 165 indicating if there

is a ring present, key 125 representing if there are more than one aromatic rings in the structure).

2.2.2. Related Works

In their SMILES2vec [44] paper, Goh et al. developed a RNN neural network architecture trained on SMILES for predicting chemical property. SMILES2vec was inspired by language translation using RNN. Goh et al. did not explicitly encode information about the grammar of SMILES. Instead, they anticipate RNN units to learn these patterns implicitly and develop intermediate features that would be useful for predicting a variety of chemical properties. RNNs, particularly those based on LSTMs [45] or GRUs [45] are effective neural network designs for learning from text data. Their effectiveness has been demonstrated in multiple works such as the Google Neural Translation Machine that uses an architecture of 8+8 layers of residual LSTM unit [46]. In SMILES2vec, they modeled sequence-to-vector predictions, where the sequence is a SMILES string, and the vector is a measured chemical property. As SMILES is a chemical language and different from spoken language, commonly-used techniques in natural language processing (NLP) research, embeddings such as Word2vec [47] cannot be directly applied. In addition, they explored the utility of adding a 1D convolutional layer between the embedding and GRU/LSTM layers. Goh et al. [33] developed Chemception, a deep CNN for the prediction of chemical properties, using only the images of 2D drawings of molecules. It was inspired by Googles Inception-ResNet [48] deep CNN for image classification. They utilized raw data in the form of 2D drawings of molecules that requires the minimal amount of chemical knowledge

to create, and investigated the viability of augmenting and possibly eliminating human-expert feature engineering in specific computational chemistry applications. Chemception was developed based on the Inception-ResNet v2 neural network architecture that combines arguably the two most important architecture advances in CNN design since the debut of AlexNet [49] in 2012 - Inception modules and deep residual learning. During the training of Chemception, additional real-time data augmentation to the image was performed so as to bolster the limited number of data available for each task. Finally, fully connected (MLP) architectures trained on fingerprint representations [14, 50, 51] are very popular in the cheminformatics community for predicting chemical properties. Although, MLP architectures trained on fingerprints are one of the earliest applications of deep learning in QSAR/QSPR modeling, they have consistently outperformed traditional ML models such as random forest and logistic regression, and DNN methods such as convolutional molecular graphs.

2.3. Method

2.3.1. Motivation

Several works [24, 52] have demonstrated the effectiveness of ensemble of different kinds of neural networks for improvement in model performance over the individual candidate neural networks. Fully connected deep neural network architectures trained on fingerprint representations [14, 50, 51] are very popular in the cheminformatics community for predicting chemical properties. Goh et al. [53] propounded the SMILES2vec architecture for treating SMILES strings as text sequences and trained recurrent neural network architectures. SMILES2vec and MLP architectures have been among the most successful

neural network architectures in predicting chemical properties. In this chapter, we harness the efficacy of these architectures and mix them into one architecture which we refer as CheMixNet. SMILES and fingerprints are the two most common representations of chemical molecules. By allowing a neural network to learn from both these representations, we could increase the generalizability and the scope of the architecture. Sequence classification on shorter texts is generally harder than on longer texts and usually has worse performance than longer texts. As SMILES2vec essentially treats the SMILES as a text with character level embedding, the performance of SMILES2vec degrades on shorter strings. Also, the performance of MLP models trained on molecular fingerprints generally varies based on the size of the molecule (performance varies based on small versus large organic molecules). CheMixNet provides a model architecture that can leverage the best of both forms of representation learned from the two inputs using appropriate neural network components for them. This provides the network with the ability to automatically assess the degree to which each representation can be leveraged for learning the given chemical property.

2.3.2. Design

Figure 4.1 illustrates our design approach for building CheMixNet models. We present three different architectures where we mix the output features learned using different types of models to learn the chemical properties from the two molecular representations as inputs- the fingerprints and the SMILES formula; hence, referred as CheMixNet. They are basically composed of two neural network branches - a sequence modeling branch that learns from the SMILES sequences using 1-D CNN and/or RNN, and a fully connected

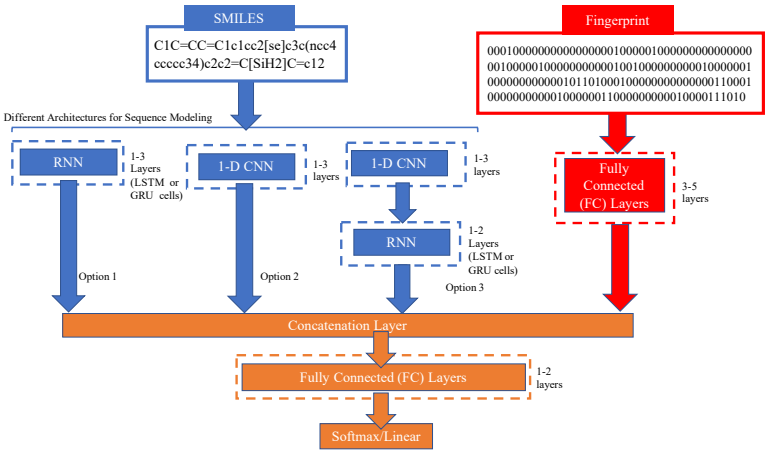


Figure 2.1. The proposed CheMixNet architecture for learning from the two molecular representations. The blue branch represents candidate neural networks for learning from SMILES sequences. Option 1 uses only LSTMs/GRUs for modeling the SMILES sequences, option 2 uses only 1-D CNNs for sequence modeling, and option 3 uses 1-D CNNs followed by LSTMs/GRUs. The fully connected (FC) branch of the model with molecular fingerprint inputs is illustrated in red. The orange part represents the fully connected layers that learn the final output from the mix of intermediate features learned by the two network branches. We exemplify the molecular fingerprint and SMILES with one representative example in this illustration.

(FC) branch that learns from the MACCS fingerprint representation. The two input representations are shown in the top of Figure 4.1. Since the SMILES representations are composed of a sequence of characters, the first network branch for learning from SMILES are composed of one of the following: 1) RNN, 2) 1-D CNN, and 3) 1-D CNN followed by RNN. Since the fingerprints are composed of bit or count vectors, the second branch for learning from them are composed of multiple layers of fully connected layers. The individual models that leverage a single input representation are composed of one of these components along with some fully connected layers at the end.

To build the mixed DNN architectures, we combine the intermediate features learned using the two branches and applied some fully-connected layers for learning the final

regression or classification output. Depending on the learning task, the output layer uses sigmoid activation for the binary classification tasks or linear activation for the regression tasks. Since we have three candidate neural networks for sequence modeling using SMILES, ChemixNet contains three neural network architectures which we refer as CNN-FC, RNN*FC, and CNN-RNN*FC where the '-' represents networks stacked in sequence and '*' represents the combination of intermediate features learned using two parallel network branches.

2.4. Data

2.4.1. Description of the Datasets

We demonstrate the effectiveness of our approach for designing DNN architectures for learning chemical properties from SMILES and fingerprints using six different datasets as shown in Table 2.1. First, Harvard CEP Dataset [14, 54] contains molecular structures and properties for 2.3 million candidate donor structures for organic photovoltaic cells. Organic Photovoltaic cells (OPVs) [55–58] are lightweight, flexible, inexpensive and more customizable compared to traditional silicon-based solar cells [59]. For a solar cell, the most important property is power conversion efficiency or the amount of electricity which can be generated due to the interaction of electron donors and acceptors, which are dependent on the HOMO values of the donor molecules. In this work, we considered highest occupied molecular orbitals (HOMO) as the target property as it determines the power conversion efficiency of a solar cell according to the Scharber model [60]. Next, we used the Tox21, HIV, ESOL, and FreeSolv (Experimental and Computed) datasets from the MoleculeNet [30, 31] benchmark repository; they involve two classification and

two regression tasks. The Tox21 dataset is an NIH- funded public database of toxicity measurements comprising of 8981 compounds on 12 different measurements ranging from stress response pathways to nuclear receptors. This dataset provides a binary classification problem of labeling molecules as either toxic or non-toxic. The FreeSolv dataset is comprised of 643 compounds that have computed and experimental hydration free energies of small-molecules ranging from 25.5 to 3.4 kcal/mol; we refer to the dataset containing experimental values as FreeSolv-Exp and the one with computed property values as FreeSolv-Comp. Hydration free energy is a physical property of the molecule which can be computed from first principles. ESOL is a dataset containing 1128 compounds with water solubility (log solubility in mols per litre) for common organic small molecules. Lastly, we evaluated the performance of CheMixNet on the HIV dataset obtained from the Drug Therapeutics Program AIDS Antiviral Screen, which measured the ability of 41,913 compounds to inhibit HIV replication in vitro. Using the curation methodology adopted by MoleculeNet, this dataset was reduced to a binary classification problem of active and inactive compounds. The original HIV dataset was very imbalanced with the minority class comprising less than 4 % of the dataset. In our work, we decided to balance the minority classes by under-sampling the majority class by randomly selecting 1,443 (size of samples from the minority class) samples. Although this led to a significant reduction in the size of the dataset, it also allowed us to investigate the viability of CheMixNet architecture on smaller datasets without significant network topology changes.

Table 2.1. Description of all the 5 datasets used to evaluate the performance of CheMixNet architectures. The original HIV dataset had 41,193 compounds but reduced to 2,886 after under-sampling.

Dataset	**Property**	**Task**	**Size**
CEP	Highest Occupied Molecular Orbital Energy	Regression	2,322,849
HIV*	Activity	Classification	2,886
Tox21	Toxicity	Classification	8,981
FreeSolv-Exp (Experimental)	Solvation Energy	Regression	643
FreeSolv-Comp (Computed)	Solvation Energy	Regression	643
ESOL	Solubility	Regression	1,128

Table 2.2. Vocab size and Maximum Input length for the datasets

Dataset	**Size of Vocabulary**	**Maximum Input Sequence Length**
CEP	23	83
HIV	54	400
Tox21	42	940
FreeSolv-Exp	32	83
FreeSolv-Comp	32	83
ESOL	33	98

2.4.2. Dataset Preparation

For the SMILES sequence, we used 1-hot encoding to convert the SMILES into a fixed length representation. The length of the sequence was determined by the length of the longest SMILES sequence in each dataset. We applied zero padding for shorter strings so that we had a uniform sequence of size equal to the maximum length for each dataset. The vocabulary size was determined by finding the number of unique characters in each dataset. Table 2.2 describes the size of vocabulary and the maximum input length for all datasets. The datasets were randomly split in the ratio of 4:1 into training and test sets. Further, the training set was split into 9:1 ratio for training and validation.

2.5. Experiments & Results

In this section, we present the experimental settings and results of the CheMixNet architectures and the comparison with other contemporary DL models on the 2.3 million CEP dataset as well as the five datasets from the MoleculeNet benchmark.

2.5.1. Experimental Settings

The DNN models were implemented using Python and Keras [61] with TensorFlow [62] as the backend. They are trained using Adam as the optimization algorithm with a mini-batch size of 32. For generating the MACCS fingerprints, we used RDKIT [63] library. Scikit-Learn [64] was used for data preprocessing and for evaluating the test set errors. All experiments are carried out using NVIDIA DIGITS DevBox with a Core i7-5930K 6 Core 3.5GHz desktop processor, 64GB DDR4 RAM, and 4 TITAN X GPUs with 12GB of memory per GPU. For our experiments, we used a learning rate of 0.001. We used the mean squared error (MSE) as the loss function for the regression tasks and used the mean absolute % error (MAPE) as the performance metric. For classification tasks, we used the binary cross-entropy as the loss function and used the area under the ROC curve (AUC) as a performance metric. Early stopping was used during training to avoid over-fitting. For the benchmark results for graph convolution networks, we used the DeepChem [65] library. For benchmarking with Chemception, there is no official public library, so we implemented the network using Keras which is also available in the CheMixNet repository [34].

We used the libraries hyperas [66] and hyperopt [67] to perform Bayesian hyperparameter search [68] to obtain the best choice of layer depth (for MLPs, 1-D convolutional and LSTM/GRU units), number of recurrent units for LSTMs/GRUs and learning rate.

Further, the Bayesian hyperparameter search was performed only for the CEP database. Once we determined the best hyperparameters for CEP, we did not change any hyperparameter except the batch size. For the CEP dataset, we used a batch size of 64; for the two classification datasets (HIV and Tox21), we used a batch size of 32; for the ESOL and FreeSolv, we used a batch size of 16.

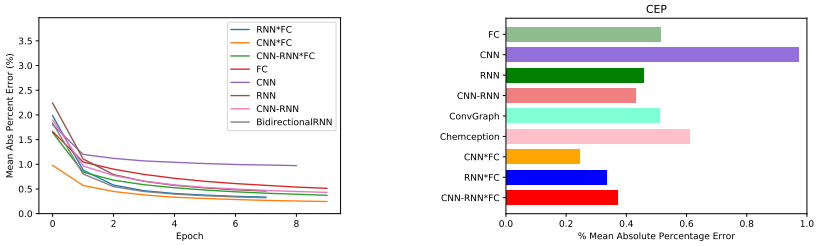
2.5.2. Results

We evaluated the effectiveness of our mixing approach for building DNN architectures to learn from both molecular representations by training the CheMixNet using six different datasets. To compare their performance, we also trained other state-of-the-art architectures for all datasets used. This includes the fully connected (FC) networks trained on the MACCS fingerprints, the two broad classes of SMILES2vec architectures - RNN and CNN-RNN, novel experimentation on SMILES using 1-D convolutions, ConvGraph and Chemception model. For the RNN and CNN-RNN architectures, we experimented using both LSTM and GRU units. As previous works [69, 70] have demonstrated the efficacy of 1-D CNN to perform effectively in text prediction without any recurrent component, we compared against 1-D CNN trained on SMILES sequences. Lastly, we compare against ConvGraph architecture that uses the molecular structure encoded as graphs as input, and Chemception architecture that uses chemical images as input.

2.5.2.1. Performance on the CEP Dataset. Figure 2.2 demonstrates the performance results of different DNN models on the CEP dataset. For the presented results, we used the MACCS fingerprints; similar metrics were observed using other types of fingerprints. The existing models trained on the SMILES generally perform better than

the models trained using only fingerprints; CNN-RNN and RNN perform significantly better than the FC model. We conjecture the difference in performance results from the difference in feature representation using SMILES and fingerprints. Since fingerprints are generated from SMILES, fingerprints are supposed to contain less information. We experimented using both LSTM and GRU for building the models composed of RNNs; for the CNN-RNN*FC, LSTM performs better than GRU while GRU performs best for CNN-RNN. Our results illustrate that the three mixed networks perform comparatively better than the existing candidate model architectures; the CNN*FC model performing significantly better. The CNN branch of CNN*FC model is composed of an embedding layer of length 32 followed by two 1-D convolutional layers with 32 filters with a kernel size of 3 (same padding and ReLU as the activation function). The FC branch is composed of four fully connected layers with 1024, 512, 256 and 64 units respectively. The final network that learns on the mixed intermediate features is composed of two layers with 64 and 1 outputs respectively. Since we perform an architecture search for each network independent of other networks, the architecture configuration for the mixed networks are different from the individual networks that leverage one input representation; hence, this is different from the current model ensemble approach where the outputs from different trained networks are aggregated to predict the output.

The derivation of fingerprints from SMILES involves simple logic and computation. However, we find that the mixing of intermediate features learned using the two network branches from the two molecular representations trained resulted in a significant gain in performance. It demonstrates the effectiveness of CheMixNet architectures in learning from multiple types of feature representations for better performance.

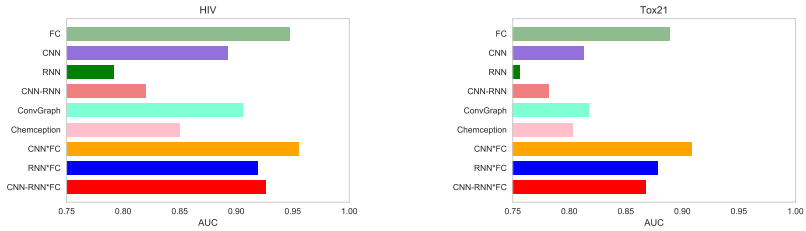


(a) Comparison of training curves

(b) MAPE comparison

Figure 2.2. Comparison of the training error curves and mean absolute percentage error on the test set for different DNN architectures on the CEP dataset. The '-' sign indicates when the networks are trained in sequence and '*' when two parallel multi-input networks (one with SMILES as input and the other with fingerprints as input) are concatenated. In these experiments, we use MACCS fingerprints - however, metrics from other fingerprints were similar. Our results demonstrate that the CNN*FC model performs the best. The three mixed networks perform comparatively better than the other state-of-the-art models. Since we use ConvGraph module from deepchem repository out of the box which does not give any information about convergence while Chemception usually takes about 100 epochs to converge, the training curve for Chemception and ConvGraph is not shown.

2.5.2.2. Performance on MoleculeNet Datasets. We further analyzed the effectiveness of using mixed networks in learning from multiple inputs by evaluating on five datasets from the MoleculeNet benchmark. Two of these datasets involves classification tasks while the rest involves regression tasks. Figures 2.3 and 2.4 illustrate the performance results of different types of model architectures on these datasets. For the two classification tasks (Figure 2.3), we observe that CNN*FC performs better than all other models. The other two mixed models CNN-RNN*FC and RNN*FC perform better than the existing models except for the FC model. FC model performs better than all other existing models on the classification tasks from the HIV and Tox21 datasets.



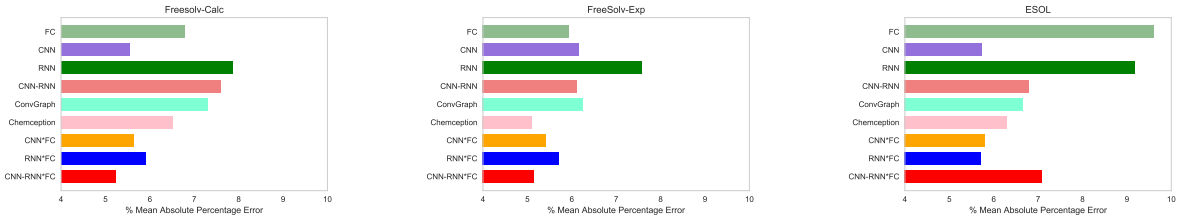
(a) HIV (b) Tox21

Figure 2.3. Performance of CheMixNet classification models against contemporary DNN models for the HIV and Tox21 datasets from MoleculeNet benchmark. CheMixNet architectures outperform the existing state-of-the-art models on both datasets. For classification tasks, higher AUC is better.

For the regression problems(Figure 2.4), we observe similar patterns- one of the mixed networks performing the best among all the networks. For the two FreeSolv datasets, CNN-RNN*FC performs the best; there is no one single best model among the existing network that works best for both these datasets. For the ESOL dataset, RNN*FC performs the best among all models; CNN model performing slightly worse. In general, we always observe benefit in performance from using mixed networks which can learn from both inputs- SMILES, and fingerprints. Since fingerprints are derived from SMILES, we conjecture the gain in performance not only comes from multiple inputs but also and more importantly from the use of different types of networks for different input representations.

2.6. Summarization

In this chapter, we present CheMixNet, the first mixed deep neural network that leverages both chemical text (SMILES) as well as molecular descriptors (MACCS fingerprints) for predicting chemical properties. Compared to existing DL models trained on single molecular representations, the proposed CheMixNet architectures perform significantly better on all the six datasets used in our study.



(a) FreeSolv (Calculated) (b) FreeSolv (Experimental) (c) ESOL

Figure 2.4. Performance of CheMixNet regression models against contemporary DNN models for the FreeSolv (calculated and experimental) and ESOL datasets from MoleculeNet benchmark. CheMixNet architectures outperform the existing state-of-the-art models on the three datasets. For regression tasks, lower MAPE is better.

The results provide proof of concept of the efficacy of using mixed input architectures for chemical property prediction. We demonstrate that by using a mixed deep learning approach, we can leverage the features of both sequence and fingerprint representations and achieve much better results, even with only a few hundred training samples. For a machine learning practitioner or researcher, the success of CheMixNet framework suggest that if several representations of data is available, an architecture that utilizes all or some of these representations is recommended.

Further, the results demonstrate that CheMixNet architectures can be generalized over a different range of chemical properties independent of the type of supervised learning tasks (classification or regression) and the type and size of datasets. The range of chemical properties predicted in our study is relevant across solar cell technology, pharmaceuticals, biotechnology, and consumer products. The CheMixNet architectures, as well as the benchmark models, are made accessible for the research community at https://github.com/paularindam/CheMixNet [34].

CHAPTER 3

Donor Property Prediction of Organic Solar Cells Using Extremely Randomized Trees

3.1. Introduction

Solar energy is a vital source of clean, versatile renewable energy and an important component in solving the worldwide energy problem [71]. Organic Photovoltaic cells (OPVs) [55–58] are lightweight, flexible, inexpensive and more customizable compared to traditional silicon-based photovoltaics [59]. However, there are challenges impeding the usage of OPVs in a commercial environment. The major issue surrounding OPVs is low power conversion efficiency of fabricated cells. Maximum cell efficiency observed in organic solar cells is currently 13.2% [72], and commercial devices usually achieve around 5-8% [73], which is much lower than silicon-based photovoltaics. The primary bottleneck in the improvement of OPV device design is complex manufacturing processes that lead to the reduction of active layer performance [74]. Traditionally, the design of a potential OPV material is dependent on conjectures from experiments, and expertise of materials scientists, followed by a laborious process of synthesis, characterization, and optimization of a prototype device.

The screening of OPV materials could be semi-automated through utilization of various modeling techniques (finite element [75, 76] to ab initio [77, 78] and molecular modeling [79]). Yosipof et al. [80] establishes the importance of data reduction and visualization

using Principle Component Analysis and Self Organizing Maps, wherein two metal oxide solar cell libraries are analyzed. Jorgensen et al. [81] describes deep generative models for predicting molecular properties, and in particular, delineates screening of OPV using molecule generation via context-free grammar VAE. Kaspi et al [82] introduces a machine learning/data mining-based decision support system PVAnalyzer for identification of interesting trends not easily observable using simple biparametric correlations, and provides scope of finding new insights into factors affecting solar cells performances. The task of screening is complicated due to the difficulty in capturing complex effects culminating from multiple local minimum configurations a polymer could adopt during the manufacturing of the active layer [83–86].

Machine learning applied to available experimental observations and theoretical simulations could potentially generate many comprehensive models with advanced predictive capabilities. This approach has been successfully applied in several materials and molecular designs [87–99] across application areas.

In this chapter, machine learning models developed using extremely randomized trees (ERTs) [100] to advance the organic monomer screening process for photovoltaic applications [101, 102] are presented. The results of ab initio simulations were combined with the cataloged description of the structural details of the monomers. The variance of structural morphology in the actual device was approximated with sets of local conformers that possibly could be created during manufacturing. Models developed in this work predict highest occupied molecular orbital (HOMO) energy of the donor monomers in the active layer of the device that is averaged across multiple configurations using Boltzmann averaging. The predicted value paired with the complementary lowest unoccupied

molecular orbital of the acceptor molecule could be used in speeding up the screening process. The proposed models outperform neural networks trained on molecular fingerprints as well as SMILES [44, 103, 104], as well as other state-of-the-art architectures such as Chemception and Molecular Graph Convolutions on both the smaller Harvard Organic Photovoltaic (HOPV) dataset as well as on a subset of the Clean Energy Project (CEP) dataset. For end-user convenience, the machine learning models were implemented as a web application at `http://info.eecs.northwestern.edu/OPVPredictor`.

3.2. Background

3.2.1. Organic Photovoltaic Cells

Among current solar cell design paradigms, organic photovoltaic cell technology is a promising technology for the inexpensive and versatile utilization of solar energy. The traditional development of new OPV materials is predominantly based on empirical intuition or experience of materials scientists. A new design idea is followed by a labor-intensive synthesis, characterization, and prototype device optimization. Hence, the problem space of OPV renewable energy research is notably complicated as the design of successful OPV materials is a multifaceted problem. The conversion of sunlight into electricity can be achieved using a solar cell and is one of the most attractive future sources of energy. Ever since the development of the first solar cells, there has been an accelerated and comprehensive exploration for cost-effective photovoltaics. OPVs are potential cost-effective and lightweight alternatives to silicon-based solar cells and could lead to the most substantial reduction of production cost. After being excited with light, firmly bound electron-hole pairs (excitons) are generated. As illustrated in Figure 3.1, an OPV works by absorbing

a photon emitted by the sun. The photon carries energy that is used to excite an electron off a donor layer, often comprised of a semiconducting polymer.

However, for the solar cell to generate electricity, the electron and hole must be separated and subsequently collected at electrodes of opposite polarity. In order to accomplish this, the exciton bond must be broken. This happens at the donor-acceptor interface, where the exciton splits into separate free electron and hole. As the charges separate further, they can reach electrodes which upon becoming charged, generate electricity as the electrons move from the cathode to the anode.

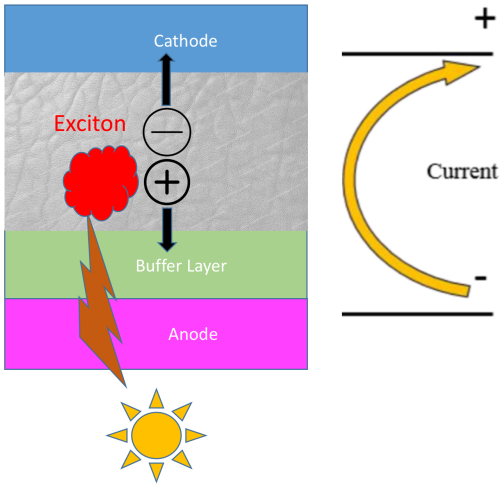


Figure 3.1. Photo-electricity generation in a bulk heterojunction Organic photovoltaic cell [3]. When the photons from the sun hit the surface of the OPV device, an electron from the donor is excited and combines with the corresponding hole at the acceptor layer to form an exciton. Electricity is generated when the exciton splits at the interface, and electrons move from the cathode to the anode.

OPVs have the advantage of combining the versatility and flexibility of plastics with photo-electronics. They can be made semi-transparent, and moldable into different forms and shapes. Researchers have even tried spray-coating OPVs on various surfaces [59].

Nonetheless, production scale small-area devices yield efficiency of only about 5% [73], with laboratory experiments yielding the highest efficiency of around 10%, and hence there is much room for improvement. Several models predict the efficiencies to reach 15% assuming the usage of state of the art materials and device architectures [105]. The conventional process for the generation of such devices is iterative and time-consuming. However, due to the labor-intensive process of generating candidates for OPVs, producing virtual screening techniques as elucidated by Pyzer-Knapp et al. [14] and our current study can potentially fasten the process considerably.

3.2.2. Scharber Model

For a solar cell, the most important property is power conversion efficiency (PCE) or the amount of electricity which can be generated due to the interaction of electron donors and acceptors. The Scharber model [60] provides a relation between the voltage V_{oc} and the energies of the HOMO and the lowest unoccupied molecular orbital (LUMO) level of the donor and acceptor molecules respectively, which in turn can be related to the power conversion efficiency (PCE), the maximum efficiency of solar cells. In the following equation, J_{sc} is the short-circuit current density, FF is electrical fill factor and P_{in} is incident-light intensity. E_{HOMO}^{Donor} and $E_{LUMO}^{Acceptor}$ indicate the HOMO and LUMO energy levels of the donor and acceptor molecules respectively.

$$V_{oc} = 1/e(|E_{HOMO}^{Donor|} - E^{|Acceptor|})_{LUMO} - 0.3V$$
$$PCE = 100 * (V_{oc} * FF * J_{sc})/P_{in}$$

3.3. Method

3.3.1. Extremely Randomized Trees

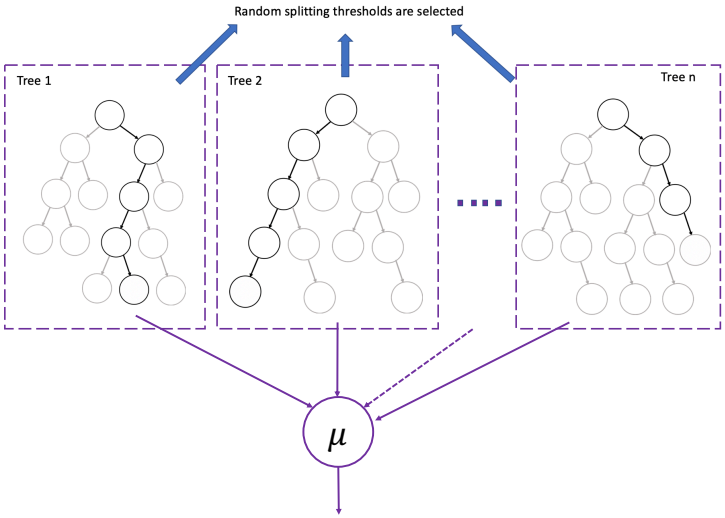


Figure 3.2. Extremely randomized trees (ERT) architecture : ERTs are a forest of decision trees where node split is selected randomly with respect to both variable index as well as variable splitting value. Results from several small trees (indicated in dashed boxes) are aggregated in ERTs. The black paths represent the decision tree path for a given data point, and the gray paths represent the decision tree paths that are not selected. The output of each individual tree is aggregated and the final predicted value is the arithmetic mean (indicated by μ).

ERTs use an ensemble of decision trees [100] in which a node split is selected completely randomly with respect to both variable index and variable splitting value. The principle behind ERTs is using several small decision trees that are individually weak learners but when aggregated in an ensemble leads to a very robust learner. ERTs are similar to other tree based ensemble algorithms such as random forests (RFs) but unlike RFs, the same training set is used for training all the trees. Further, ERTs split a node based on both variable index and variable splitting value while random forests only splits by variable

value. This makes ERTs both more computationally efficient than RFs and generalizable. Figure 3.2 illustrates the working of ERTs by aggregating results from several smaller trees.

One of the reasons for choosing a tree-based ensemble approach is that it is possible to develop an effective model even on a very small dataset. For other algorithms such as neural networks, there are many more parameters to learn and therefore, it requires a higher amount of data. Furthermore, decision-tree based methods rank the top features that allows us to interpret the impact of these features on the output attribute. In other algorithms employing neural networks, it is difficult to draw connection of the output attribute with the original features. One of the motivations of this work is to develop an interpretable model that can be used by molecular scientists.

To evaluate the validity of ERTs to scale to other datasets, we experimented on a subset of the Harvard CEP Dataset [14] which contains DFT-calculated molecular structures and properties for many candidate donor structures for organic photovoltaic cells. The CEP is a virtual high-throughput discovery and design effort for the next generation of plastic solar cell materials. It studies many candidate structures to identify suitable compounds for the harvesting of renewable energy from the sun and for other organic electronic applications. To establish the generalization of the models for larger datasets, we scraped a portion of the CEP database available. For scraping, we used the python libraries selenium [106] and beautiful soup [107]. This dataset is made available in the supplementary material. We restricted our extraction to 22,179 data points as the online CEP database had restrictions in place preventing automatic web-extraction of the entire database.

3.3.2. Datasets

The HOPV dataset [108] used in this work is a collection of photovoltaic measurements for a diverse set of 350 organic donor compounds generated by extensively searching the literature. In our experiments, the dataset was reduced to 344 molecules after removing redundant isomeric samples [109]. The dataset provides density functional theory (DFT) calculations of HOMO energy values for four functionals B3LYP, BP86, PBE and M06 using the basis set def2-SVP [110]. We get the expected values for HOMO values across all conformers by calculating the boltzmann average. Each molecule in the HOPV dataset is represented by a subset of 3-18 conformers obtained at kT, where k is the Boltzmann constant and T is the temperature of the OPV device. The global minimum (T = 0K) structures used for prediction of HOMO energies are far from the donor molecule structures in real OPV devices, after various manufacturing steps. We observe from Figure 3.3 that the PCE of the OPV device and the HOMO energy values are correlated with each other. We abstained from building models on the experimental values as HOMO values were missing for many molecules, and manufacturing information was not provided.

The band gap of the processed organic layer (made up of donors, acceptors, and other additives) would be altered from their global minimum value due to the shift of molecules from their ideal configuration. The degree of alteration would depend on the exact routine used in manufacturing, and is hard to predict. The boltzmann averaging is an attempt to account for the effect of structural variation in the experimental device. This is because different conformers of the same molecule occur in real OPV devices, and hence HOMO energies averaged over all conformers into the predictive model is expected to improve the relevance of the predicted HOMO values to the performance of the actual device.

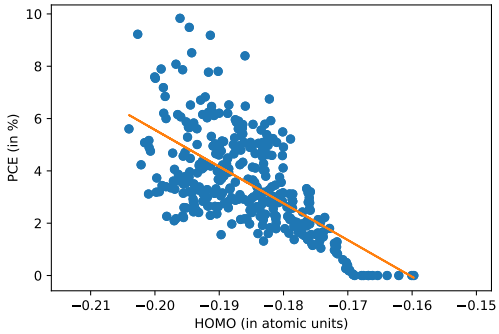
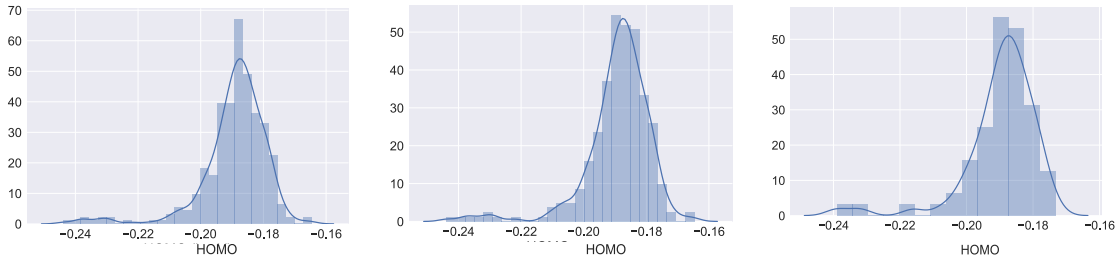


Figure 3.3. The scatter-plot (with line of best fit) demonstrates the linear relationship between PCE of the device and HOMO values of the donor compound. The boltzmann average of the HOMO values for each conformer is used to determine the HOMO for a given donor.



(a) (b) (c)

Figure 3.4. Distribution of the datasets : (a) entire HOPV dataset, (b) training set, and (c) held-out test set. All the HOMO values are in atomic units (a.u.). 1 atomic unit is equal to 27.21 eV.

3.3.3. Data Mining

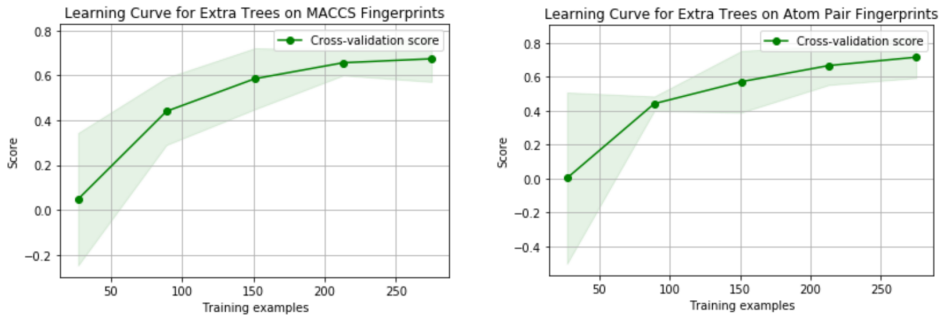
For both the datasets, the original data was divided into training and test subsets. Figure 3.4 illustrates the distribution of the HOMO values across the complete HOPV dataset, the training and test sets. The dataset is split into training and test subsets with 80% and 20% of the data points respectively. We use stratified shuffle splitting to ensure similar distribution across the training and test set. The HOPV dataset provided DFT calculations for 4 functionals : PBE, B3LYP, BP86 and M06. In this work, we restricted

ourselves to PBE calculations. Further, we found that all the other functionals can be expressed as a linear transformation of the PBE functional values.

Two fingerprint representations - MACCS and Atom Pair were used for generating features [111–116]. For Atom Pair fingerprints, we initially calculated the original unhashed count vector of length 4 million for all the molecules using RDKit. After that, features that are invariant across the entire dataset were removed. This led to the reduction of the length of the unfolded fingerprint from 4 million to 2696 . The uncompressed MACCS fingerprint was only 166 bits long, and hence no feature reduction or transformation was performed. We did not use 1024 bit compressed fingerprint representation for Atom Pair as the original meaning of the fingerprint would be lost.

The fingerprints were prepared from their simplified molecular-input line-entry system (SMILES) [117] formulae using RDKit Python Library [118]. SMILES is a form of line notation for the chemical structure of molecules, and considered a versatile system. Molecule editors can generate 2D and 3D models from the line notation. The HOPV dataset provides canonical Standard SMILES implementations both in standard and shortened format.

Extensive grid search was performed across hyper-parameters to discover the model architecture with the least mean absolute error for 5-fold cross-validation. This model was chosen and trained on the entire training dataset.



(a) (b)

Figure 3.5. Learning curves for the cross-validated ERT models across different set of training examples for the MACCS and Atom Pair Fingerprints. The goodness of prediction (Q^2) is used as the score.

3.4. Results & Discussion

3.4.1. Experimental Results

In this work, we provide a framework for reducing the design space by screening new donor candidates using machine learning models developed on the HOPV dataset. Although both donors and acceptors are essential for an OPV application, the current work is restricted to donors as there are only a small number of known acceptors [119, 120] compared to hundreds of thousands of potential donor molecules. Therefore, developing a machine learning-based screening solution for donor molecules would lead to the identification of OPV devices with high PCE.

Figure 3.5 demonstrates the learning curve of the cross-validated ERT models across different set of training examples. The learning curves help demonstrate the increase of the learning capacity of the model as the dataset is increased. Further, the variance of the cross-validated models (indicated by the shaded green band surrounding the corresponding curve) decreases as the number of training examples increase.

Table 3.1. Comparison of performance of ERT models with other algorithms for the HOPV dataset

Algorithm	**Feature**	**% MAE**	**RMSE**	Q^2
AdaBoost	Molecular Fingerprint (MACCS)	2.6443	0.0061	0.1670
AdaBoost	Molecular Fingerprint (AtomPair)	2.5395	0.0058	0.2269
XGBoost	Molecular Fingerprint (MACCS)	2.0472	0.0057	0.7277
XGBoost	Molecular Fingerprint (AtomPair)	2.0141	0.0057	0.7263
Bagging	Molecular Fingerprint (MACCS)	2.6162	0.0063	0.1098
Bagging	Molecular Fingerprint (AtomPair)	2.4500	0.0058	0.2503
Random Forest	Molecular Fingerprint (MACCS)	2.0977	0.0054	0.4982
Random Forest	Molecular Fingerprint (AtomPair)	2.0589	0.0053	0.5169
ERTs	**Molecular Fingerprint (MACCS)**	**1.9703**	**0.0057**	**0.7390**
ERTs	**Molecular Fingerprint (AtomPair)**	**1.9100**	**0.0056**	**0.7427**
FC	Molecular Fingerprint (MACCS)	3.6850	0.0084	-0.5906
FC	Molecular Fingerprint (AtomPair)	3.5135	0.0078	-0.3975
CNN	SMILES	3.2536	0.0072	-0.1885
RNN	SMILES	2.6240	0.0062	0.1200
CNN-RNN	SMILES	2.6443	0.0061	0.1670
ConvGraph	Molecular Graphs	2.8170	0.0079	0.1082
Chemception	Molecule Image	3.2738	0.0079	-0.4089

To compare their performance, we also trained other state-of-the-art architectures for all datasets used. This includes a fully connected (FC) network trained on the fingerprint representations. Further, we also compare against 1-D CNN, RNN and CNN-RNN architectures trained on SMILES as recent papers have demonstrated their superiority over FC methods [44, 103, 104]. Lastly, we compare against other state of the art neural networks used in molecular informatics such as ConvGraph and Chemception. While the ConvGraph architecture uses the molecular structure encoded as graphs as input and then performs graph convolutions, Chemception architecture [33], based on the Inception architecture for image classification [121], directly develops a very deep neural network model by training directly on images of molecules. Bagging, RandomForest, ERTs and AdaBoost algorithms were implemented using Scikit-Learn Python Library [122]. The

Table 3.2. Performance metrics of the randomization tests performed using the MACCS and AtomPair fingerprints as features

Features	**Model**	**% MAE**	**RMSE**	Q^2
MACCS	y-Randomization	4.6036	0.0117	-2.1476
	Pseudo-Descriptors	6.5617	0.0167	-5.3666
Atom Pair	y-Randomization	3.3981	0.0083	-0.5600
	Pseudo-Descriptors	5.5822	0.0147	-3.9450

XGBoost package [123] was utilized for creating the xgboost model. The FC, CNN, RNN, CNN-RNN and Chemception models were implemented using Keras [61] with Tensorflow [62] backend. The ConvGraph was implemented using DeepChem library [65].

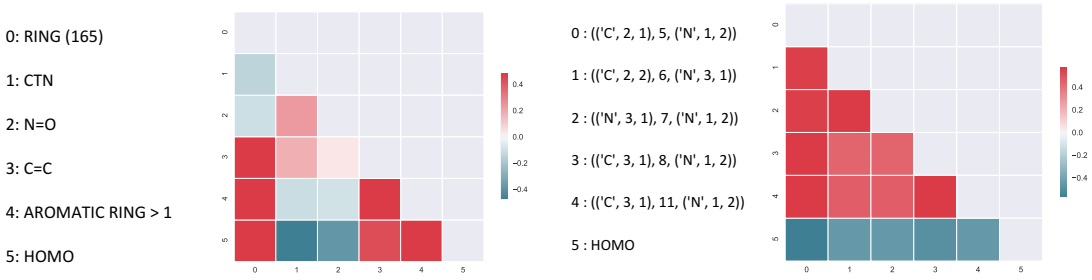
In Table 5.1, we present the results of the experiments across all the models for the HOPV dataset. We present the % Mean Absolute Error (MAE), Root Mean Squarer Error (RMSE) and goodness of prediction (Q^2). We can observe the superiority of ERTs for both the MACCS and Atom Pair fingerprints over the other models. ERTs trained on MACCS and Atom Pair had a mean absolute percentage error (% MAE) of 1.91% and 1.97%. The RNN, CNN and CNN-RNNs trained on the SMILES had % MAE between 2.62% and 3.25%. Convolutional Graphs had % MAE of 2.82 % and all other methods based on deep neural networks had even higher % MAE. Two ensemble tree based algorithms XGBoost and Random Forest outperform all other methods except ERTs. Even other ensemble tree-based algorithms such as AdaBoost and Bagging perform relatively well and at par with the best neural network based methods (RNNs and CNN-RNNs). It must be noted that although ERT models outperform RF models based on % MAE (lower %MAE) and Q^2 (much higher Q^2), RF models have slightly lower RMSE.

In Table 3.2, the results of the randomization tests such as y-Randomization and pseudo-Descriptor tests are delineated. y-Randomization (also known as y-scrambling

or response randomization) is a form of a permutation test, where the values of the response variable are randomly ascribed to different compounds, while the descriptors values are left intact. In the pseudo-descriptors test, the descriptors are replaced by random numbers that are also subsequently used to train the models. In our case as the features in fingerprints are bit vectors, we generate random bit strings for features. A comparison across the performance metrics such as % MAE, RMSE and Q^2 of the ERT models between the original dataset (in Table 5.1) and the randomization tests (in Table 3.2) demonstrates that our proposed models perform much better than models based on random input features (pseudo-Descriptors) or labels (y-Randomization).

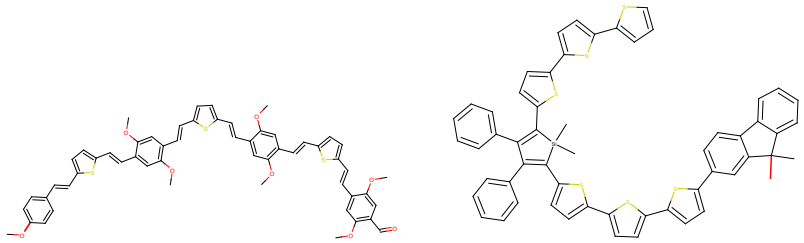
3.4.2. Correlation of fingerprint features

We wanted to explore the correlation between the most important features for our model for understanding their impact on the HOMO value. Figure 3.6 depicts the correlation matrices for top 5 features important for MACCS and Atom Pair Fingerprints, as they perform best across all the fingerprints. We restricted to top 5 features as the contribution of other features was very close to 0. The length of MACCS fingerprints is 166, which is much shorter compared to other fingerprints, and is least affected by the curse of dimensionality. The correlation plots demonstrate that presence of any ring (Feature 0), presence of a C=C double bond (Feature 3) and presence of an aromatic ring (Feature 4) is positively correlated with HOMO value, whereas a C≡N triple bond (Feature 1) and a N=O double bond (Feature 2) is negatively correlated with HOMO value. Further, the correlation plot illustrates that presence of any ring, the presence of C=C bond and presence of an aromatic ring are strongly positively correlated with each other and hence



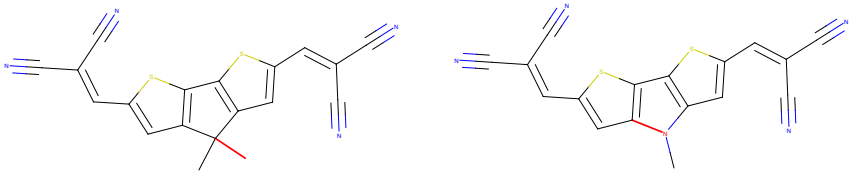
(a) MACCS (b) AtomPair

Figure 3.6. Correlation across the top 5 features and HOMO for MACCS and AtomPair fingerprints



(a) (b)

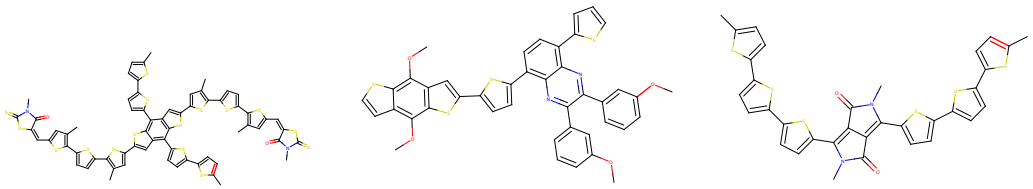
Figure 3.7. Specimen donor molecules with the highest HOMO



(a) (b)

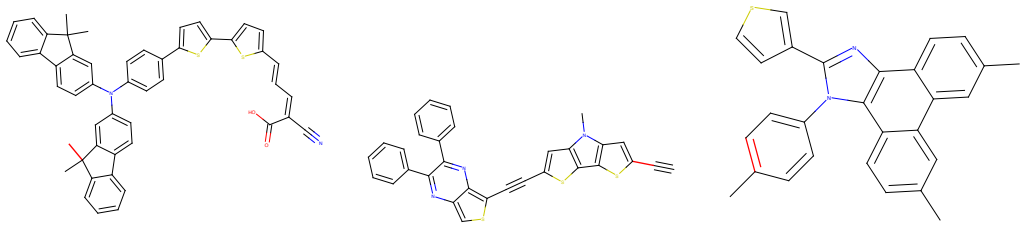
Figure 3.8. Specimen donor molecules with the lowest HOMO

we can conclude that these features often co-occur together in compounds with high HOMO value. Similarly, C≡N triple bond and N=O double bond have a weak positive correlation with each other, and their co-occurrence together leads to a compound with low HOMO value.



(a) (b) (c)

Figure 3.9. Best predicted structures based on prediction by both MACCS and Atom Pair Fingerprints



(a) (b) (c)

Figure 3.10. Worst Predicted Structures based on prediction by both MACCS and Atom Pair Fingerprints

Figure 3.7 depicts two compounds with the highest HOMO value, and the abundance of rings including aromatic rings correspond to our observation from the correlation plots. Figure 3.8 illustrates two compounds from the HOPV dataset with the lowest HOMO value, and the presence and abundance of C≡N triple bond and N=O double bond are per our expectation based on correlation values. Although all compounds in the HOPV dataset had aromatic rings as the fingerprints are count vectors and not bit vectors, it demonstrates that the number of rings positively correlate to higher HOMO value rather than the presence or absence of rings.

Figure 3.9 depicts the best-predicted structures from the dataset with respect to predictions based on both atom pair and MACCS fingerprints. All the compounds that are

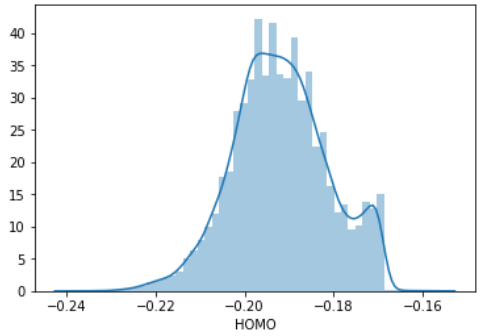


Figure 3.11. Distribution of the CEP subset. All the HOMO values are in atomic units (a.u.). 1 atomic unit is equal to 27.21 eV.

predicted well have many aromatic rings, in agreement to our models as the number of rings and the number of aromatic rings are essential features. On the contrary in Figure 3.10, the compounds have fewer aromatic rings, and also have many features that are not part of the important features in the extremely randomized tree model. This makes it difficult to accurately predict the HOMO value. Although in this work, the predicted feature is HOMO and not PCE, the demonstrated dependence of HOMO and PCE (via the Scharber model as well as illustrated in Figure 3.3 implies that PCE values are correlated directly to HOMO.

3.4.3. Generalization on Larger Dataset

We explored ERTs on the larger dataset of 22,179 molecules extracted from the Harvard CEP Database. We present the distribution of the HOMO values of the larger dataset in Figure 3.11. The reported HOMO values in the CEP dataset are an aggregate across several functionals. Table 3.3 compares the performance of the ERT models with other algorithms. As this dataset is much larger compared to the 350 molecule HOPV dataset, some deep neural methods such as convolutional graphs expectedly perform comparable

Table 3.3. Comparison of extremely randomized tree models with other algorithms for the 22,179 molecule CEP dataset

Algorithm	**Feature**	**% MAE**	**RMSE**	Q^2
AdaBoost	MACCS	2.0349	0.1284	0.7210
AdaBoost	AtomPair	2.0170	0.1272	0.7261
XGBoost	MACCS	0.9430	0.0611	0.9558
XGBoost	AtomPair	0.9378	0.0622	0.9523
Bagging	MACCS	1.6434	0.107	0.8065
Bagging	AtomPair	1.6418	0.1076	0.8551
Random Forest	MACCS	1.4331	0.0946	0.8864
Random Forest	AtomPair	1.4654	0.0967	0.8819
ERTs	**MACCS**	**0.8991**	**0.0598**	**0.9572**
ERTs	**AtomPair**	**0.8696**	**0.0584**	**0.9604**
FC	MACCS	1.6444	0.1070	0.8065
FC	AtomPair	1.6226	0.1058	0.8107
CNN	SMILES	0.7804	0.0521	0.9673
RNN	SMILES	0.7815	0.0527	0.9663
CNN-RNN	SMILES	0.7786	0.0529	0.9667
ConvGraph	Molecular Graphs	0.9104	0.0519	0.9619
Chemception	Molecule Image	1.4681	0.0974	0.8762

to the ERTs, and SMILES-based models slightly outperform the ERT models. As the dataset is larger, we increased the number of trees in our model to 200.

Although the Scharber model is simplistic to account for all the complex physics of an OPV explicitly, it nonetheless provides a valuable indication of the inherent promise of a candidate compound. Further, as the HOPV dataset was small, the web application must be used with caution. Due to the low mean absolute percentage error (% MAE), it will have high precision for compounds that are similar to those in the HOPV dataset. For instance, the HOPV dataset has only 3 compounds that have Selenium in the donor molecule.

3.5. Summarization

A methodology for predicting properties using fingerprints of donor molecules is presented. The elegance of an ensemble based regression technique such as ERTs lies in the fact that it minimizes the need for feature reduction or normalization. In particular, ERTs are generalizable and less prone to overfitting which is essential while learning from a small dataset. Further, ERTs are easily interpretable - a desired trait for further understanding of which features are most important for the predicted property of a given monomer. One of the goals of machine learning models is reusability. In the proposed work, although the models were trained using the PBE functional values, we ascertained that HOMO values of other functionals namely B3LYP, BP86, and M06 could be expressed as a linear transformation of their corresponding values for B3LYP functionals. Hence, the models developed for PBE can be extended to predict for other functionals. For the smaller OPV dataset, ERT models achieve better performance than other methods -both tree-based as well as those based on neural network. Further, we evaluated ERTs on the larger dataset and it performed almost at par with CNN or RNN-based neural networks trained on SMILES. We also provide a web application where users can receive the predicted HOMO values for the chemical compound of the donor as well as V_{oc} of the donor-acceptor combination for a given acceptor.

This work reveals the potential of integration of feature manipulation combined with extensive grid search on a small experiment-theory calibrated dataset of organic photovoltaic donors. Our system allows researchers to get an estimate of the HOMO energy values of donor compounds used in OPV applications, and motivate the development of an inexpensive photovoltaic solution.

CHAPTER 4

Transfer Learning Using Ensemble Neural Networks for Organic Photo-voltaic Applications

4.1. Introduction

Based on current statistics, energy consumption had increased from 238 Exajoules (EJ) in 1972 to 464 EJ in 2004. A further 65% increase is projected by the year 2030 [124]. Sustained usage of fossil fuels leads to irreversible changes to the planet with sea level rising from 1.7 to 3.2 mm per year and ocean temperatures increasing [124, 125]. It is imperative to search for versatile and cost-efficient clean energy solutions to prevent further irreversible damage. One limitation with renewable energy is that it is difficult to generate the quantities of electricity that are as large as those produced by traditional fossil-fuel generators [126]. Wind and hydro-energy solutions require expensive installations [127] and maintenance, which requires large government grants, and are dependent on weather and climate conditions [128]. Moreover, most renewable energy technology is new and has enormous capital costs compared to traditional fossil fuels. Solar energy provides a more cost-effective solution with faster installation, and more predictive energy outputs based on the Bureau of Meteorology and National Aeronautics and Space (NASA) reports [129]. Although inorganic silicon-based solar energy systems are currently more conventional, organic or plastic photovoltaic (OPV) [130] technology has become very popular because of its flexibility. Organic or Plastic Technology is very versatile as demonstrated by how

plastics in consumer goods can be made very hard and durable, or very light or transparent as dictated by needs [131]. Further, manufacturing costs are lower for organic solar cells compared to silicon-based materials due to the ease of device manufacturing, and lower cost of organic components compared to silicon [132].

However, the main bottleneck in the deployment of organic solar cells is that the search for candidate chemical compounds for creating organic solar cells is very time consuming [133]; it can take up to thousands of hours of laboratory analysis. For a solar cell, the most important property is power conversion efficiency (PCE) or the percentage of electricity which can be generated due to the interaction of electron donors and acceptors after absorption of energy from the sun. The PCE is dependent on the highest occupied molecular orbital (HOMO) energy of the donor and the lowest unoccupied molecular orbital (LUMO) energy of the acceptor molecule [60]. However, as the LUMO values across known acceptors do not vary much, and only a few acceptor molecules exist, predicting HOMO values of donor molecules can give us estimates of PCE when those donors are used in solar cells.

There are two main issues with the current practice of building predictive models using machine learning (ML) techniques. First, these predictive models are built using a single representation of the molecular structure - line notations [36] such as SMILES or InChI, molecular fingerprints [14] or molecular graphs [32]. Line notations are increasingly becoming popular for use in ML models as molecular fingerprints are difficult to interpret and models trained on molecular graphs usually perform worse [44]. However, these approaches have restricted themselves to only one type of line notation – either SMILES or InChI as input representations for predictive models. This limits the information that

can be harnessed from these representations as SMILES and InChI express the molecular structure in very different ways. SMILES defines the chemical bond types present in the molecular structure from which one can infer the protonation while InChI serves the opposite purpose- it defines the protonation from which one can infer the chemical bond types present in the molecular structure. Also, SMILES was designed to be read and written by humans whereas InChi was intended to ignore tautomeric form and be more consistent. Since the two line representations are distinct in their properties, a predictive model can benefit from the use of both of them. Second, most of the datasets, especially, experimental datasets are limited in size. Hence, current ML models are either built using publicly available large DFT-computed datasets such as the Harvard CEP dataset [14, 54] or other limited experimental or DFT-computed datasets which are relatively smaller in size and hence, the model cannot learn the required data representation for making robust predictions. In this work, our goal is to leverage together larger DFT-computed datasets with relatively smaller datasets such as experimental observations and combine both types of line representations- SMILES and InChI – to build more robust predictive models for predicting HOMO values for donor candidates for OPV.

We present an ensemble deep neural network architecture, called SINet, which leverages both the SMILES and InChI molecular representations to learn to predict the HOMO values, and leverage transfer learning from large datasets to build more robust predictive models for a relatively smaller dataset. SINet is composed of two identical branches for both types of inputs; each branch consists of 1-D CNN layers followed by LSTM layers. The features learned by the two branches from the two input representations are combined and fed into a fully connected network for predicting the regression output

of HOMO value. The deep neural network architecture of SINet enables us to perform transfer learning from a large dataset to relatively smaller dataset in a similar domain. Transfer learning has already been adopted in the fields of computer vision, natural language processing and other application domains [134, 135].

Our source dataset for transfer learning is the Harvard CEP dataset [14, 54] which contains molecular structures and properties for 2.3 million candidate donor structures for OPV. For the target dataset, we leverage DFT-computed and experimental values of 350 and 243 molecules respectively, from the HOPV [108]. Our results demonstrate significant performance improvement from the use of both types of inputs- the MAPE drops from 0.972% and 0.457% using SMILES and InChI respectively, to 0.213% when using them together using the SINet architecture. We also find significant benefit from using transfer learning from Harvard CEP to the HOPV datasets. The MAPE for experimental and DFT-computed datasets from HOPV drops from 2.782% to 1.513% and 2.118% to 1.478%, respectively. Since the model is first trained on a large dataset, it learns the required set of features from the input data representation, and this helps in learning the similar features present in the smaller target dataset, on which the model is fine-tuned. Our results demonstrate significant benefit from the use of both types of input representations as well as from transfer learning from a larger dataset. It showcases that leveraging machine learning with computational and experimental chemistry can play an essential role in the expedition of a systematic design of high-efficiency OPV materials, and holds significant promise as a potential solution to future energy needs.

4.2. Method

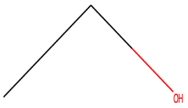
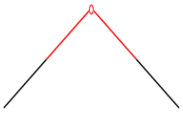
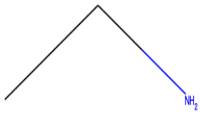
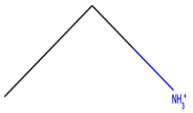
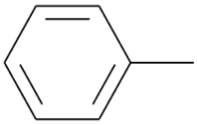
In this section, we discuss the InChI representation, present the source and target organic photovoltaic datasets used in our experiments, discuss the preprocessing of the SMILES and InChI strings and propound our methodology.

4.2.1. InChI

The IUPAC International Chemical Identifier (InChI) [136] was developed by IUPAC and NIST (National Institute of Standards and Technology). It is a textual identifier for chemical substances which provides a standard way to encode molecular information. Every InChI string starts with the string "InChI=" followed by the version number and letter S in the case of standardized InChIs. Rest of the InChI string is structured as a sequence of layers and sub-layers. Each layer provides a specific type of information, and are separated by "/". The InChI algorithm transforms the structural information of the molecule into a unique InChI identifier in a three-step process. The first step is normalization which removes redundant information. This is followed by canonicalization that generates a unique number label for each atom. The last step is serialization that produces a string of characters.

SMILES and InChI are distinct notations. SMILES defines the bond types from which one can infer protonation, while InChI defines protonation from which one can infer the bond types. SMILES was designed to be read and written by humans and is therefore relatively straightforward to read, provided the user knows a few basic principles of the format. InChI, is comparatively less readable, is intended to ignore tautomeric

Table 4.1. Examples of set of similar chemical compounds with their corresponding SMILES and InChI notations with explanation

Compound 1	Compound 2	Line Notations (with explanation)
 Ethanol	 Dimethyl Ether	• SMILES: CCO and COC • InChI: InChI=1S/C2H6O/c1-2-3/h3H,2H2,1H3 and InChI=1S/C2H6O/c1-3-2/h1-2H3 • "C2H6O" means that the first and second atoms (1 and 2) are C atoms and the third (3) is an O atom. The connectivity is 1-2-3 for ethanol and 1-3-2 for dimethyl ether. For ethanol atom 3 has 1 H atom, atom 2 has 2 H atoms, and atom 1 has 3 H atoms. For dimethyl ether atom 1-2 have 3 H atoms, while atom 3 has none.
 Ethylamine	 Ethylammonium	• SMILES: CCN and CC[NH3+] • InChI=1S/C2H7N/c1-2-3/h2-3H2,1H3 and InChI=1S/C2H7N/c1-2-3/h2-3H2,1H3/p+1 • For the InChI notation, the protonation (h2-3H2,1H3) is identical in both cases an corresponds to ethylamine. For ethylammonium "p+1" indicates that an extra proton is added.
 Benzene	 Toluene	• SMILES: c1ccccc1 and Cc1ccccc1 • InChI=1S/C6H6/c1-2-4-6-5-3-1/h1-6H and InChI=1S/C7H8/c1-7-5-3-2-4-6-7/h2-6H,1H3 • For the SMILES notation, in the case of benzene atom 1 is connected to both atom 2 and atom 6, i.e. a ring is formed. "1" is a label and does not refer to atom number 1 (see toluene). A lower case "c" is used to indicate aromatic carbons, meaning they should be singly protonated. For toluene, the methyl group is bonded to atom number 2, which is also bonded to atom number 7. For the InChI notation, in the case of benzene aromaticity is inferred from the fact that all 6 carbon has 1 H atom (h1-6H). For toluene, the methyl group is bonded to atom number 7, which is also bonded to atom number 6.

form and is consistent. Table 4.1 illustrates the distinction between SMILES and InChI representations.

4.2.2. Datasets

The source dataset for transfer learning is the Harvard CEP Dataset [14, 54] which contains molecular structures and properties for 2.3 million candidate donor structures for organic photovoltaic cells. For a solar cell, the most important property is power conversion efficiency or the amount of electricity which can be generated due to the interaction of electron donors and acceptors, which are dependent on the HOMO values of the donor molecules. In this work, we considered the highest occupied molecular orbitals (HOMO) as the target property as it determines the power conversion efficiency of a solar cell according to the Scharber model [60].

The target dataset was the Harvard Organic Photovoltaic (HOPV) dataset [108] which is a collection of photovoltaic measurements for a diverse set of 350 organic molecules generated by extensively searching the literature. Of these, experimental values were available for 243 molecules and calculated values using density functional theory (DFT) were available for 344 molecules. In our experiments, the DFT-computed values in the HOPV dataset were reduced to 344 molecules after removing redundant isomeric samples [109]. We used both the experimental and calculated datasets as target datasets for transfer learning. The HOPV dataset contains density functional theory (DFT) calculations for four functionals B3LYP, BP86, PBE and M06 using the basis set def2-SVP [110]. We used B3LYP functional values as it is the most popular functional for HOMO value calculations. Further, HOMO values across all conformers were Boltzmann-weight averaged [137].

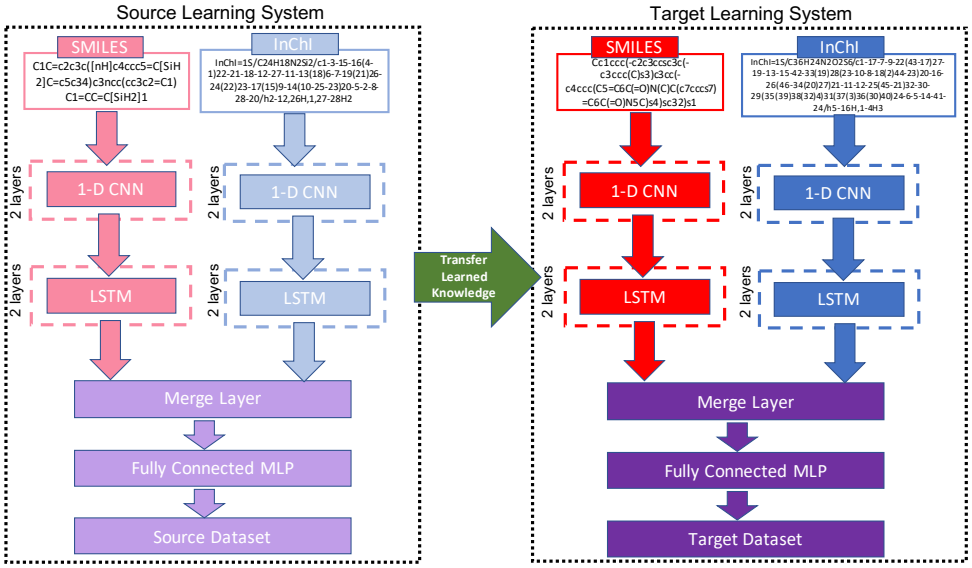


Figure 4.1. The proposed SINet architecture for learning from the two text-based molecular representations - SMILES and InChI. The left side (represented by faded colors) represents the learning from the source dataset while the right side (represented by darker colors) represents the learning for the target dataset. For both the learning systems, the red branch represents the network for sequence modeling from SMILES while the blue branch represents the network for sequence modeling from InChI. The purple part represents the fully connected layers that learn the final output from the combination of features learned by the two network branches. We exemplify the SMILES and InChI with one representative example in this illustration from both the source as well as target datasets.

4.2.3. Preprocessing

For both the SMILES and InChI sequences, one-hot encoding was performed separately to convert them into fixed length representations. The sequence lengths were calculated using the length of the longest SMILES and InChI sequences in the dataset respectively. To maintain a uniform sequence size, shorter strings were padded with zeros. Similar to SMILES2vec, vocabulary size was equal to the number of unique characters. The sequence lengths are 82 and 162 respectively for SMILES and InChI input representations.

4.2.4. SINet

Figure 4.1 illustrates the proposed approach for performing transfer learning from the Harvard CEP dataset to the two smaller target datasets in the HOPV dataset. The deep neural network architecture used for the task consists of two branches for the two types of the input representations of SMILES and InChI. The SMILES input vector has a length of 82 while the InChI input has 162 values. Both branches have the same network configuration. Each branch is composed of a 1-D CNN followed by an LSTM network. The 1-D CNN is composed of two layers with 32 filters each; the filter size used in each layer is 3 and same padding for the inputs and output. The convolutional layers are followed by max pooling with a pool size of 2. There was no significant difference with other types of pooling and other pooling sizes. The output of the 1-D CNN is fed into the LSTM network which is composed of 2 layers having 64 units each. Finally, the outputs from both branches are concatenated into the merge layer and fed into a fully connected network which is composed of a penultimate dense layer with 64 units and the final layer that gives the HOMO value as the regression output. Since the network architecture leverages both SMILES and InChI molecular representations, we refer to it as SINet.

For transfer learning, first, we train a model on the source dataset of Harvard CEP from scratch (by initializing the model parameters from scratch before training). While being trained on the large dataset, the model learns a rich set of feature representations present in the large training data which is useful for making predictions in the source domain. Next, for using transfer learning, we can follow one of the two techniques. Either, the same trained model can be fine-tuned by training on the target dataset, or we can initialize a new model using the model parameters from the model trained on

the source data and then fine-tune it on the target data. In this work, we use the first approach as the target dataset is very small, and we wanted to harness the source dataset as much as possible. In the case of transfer learning, rather than learning all the feature representations present in the input data from scratch, the model already knows the input data distribution from the source dataset and only fine-tunes its parameters to adapt to the target dataset.

4.3. Experiments & Results

In this section, we present the experimental settings and results of the ensemble SINet architecture including the impact of transfer learning for performance gain on the smaller HOPV datasets.

4.3.1. Experimental Settings

The models were implemented using Python and Keras [61] with TensorFlow [62] as the backend. We used Adam as the optimization algorithm with a mini-batch size of 32. For generating the InChI fingerprints for the CEP dataset, we used RDKIT [63] library to generate InChI from the molecules. Scikit-Learn [64] was used for data preprocessing and for evaluating the test set errors. All experiments are carried out using NVIDIA DIGITS DevBox with a Core i7-5930K 6 Core 3.5GHz desktop processor, 64GB DDR4 RAM, and 4 TITAN X GPUs with 12GB of memory per GPU. We performed extensive hyperparameter search as well as architecture search for SINet. For our experiments, we used a learning rate of 0.001. We used the mean squared error (MSE) as the loss function and used the mean absolute % error (MAPE) as the performance metric. Early stopping

was used during training to avoid over-fitting. For our experiments, we split each dataset into 70-20-10 ratio for training, test and validation sets; we used the same split for all experiments of each dataset. Stratified shuffling was used to ensure that the distribution of HOMO values for all the 3 subsets was similar.

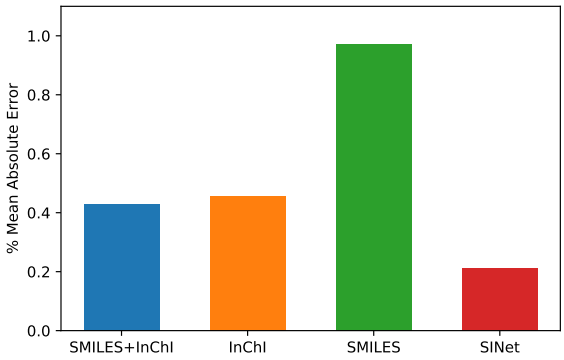
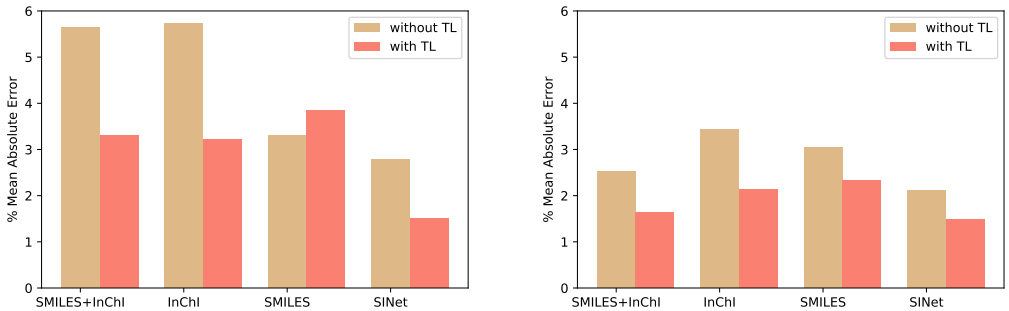


Figure 4.2. Mean Absolute Error Percentage for the CEP Dataset (source dataset)



(a) Experimental (b) DFT

Figure 4.3. Mean Absolute Error Percentage for the OPV Datasets without and with Transfer Learning (TL)

4.3.2. Impact of Leveraging SMILES & InChi

First, we explored the performance of using different types of input representations and their combinations on the source and the target datasets. On the source dataset of Harvard CEP, we observe that the MAPE decreased to 0.213% while using the SINet model as shown in Figure 4.2. In contrast, while using the individual input representations with the individual branch of SINet, the MAPE values were 0.972% and 0.457% using SMILES and InChi. This was also true for the target datasets of experimental and DFT-computed values from HOPV. We conjecture the prediction mainly improves while leveraging multiple molecular representations because the two line notations- SMILES and InChI differ in the representation and detail; hence, the model can learn different feature representation from the two input representation, leading to better performance.

Furthermore, we experimented with simply combining the two input representations- SMILES and InChI into a single input vector (represented as SMILES + InChI in Figure 4.2), before feeding them into a branch of SINet, the MAPE, in this case, was 0.430% which is lower than while using single input. However, there was no benefit from simply combining the two input representations into a single vector in the case of experimental target dataset. We surmise that this could be because the two line notations encode different representations with varying lengths for different compounds, and a concatenation of the representations was not sufficient for learning both notations. Our results recommend that a better way to incorporate multiple input representations such as SMILES and InChI, in this case, is to design the deep neural network to have different model components to handle each of them before the learned features can be combined to make the final output as in the case of SINet.

In addition, we can observe the impact of training data size on the prediction performance; the prediction error of SINet on Harvard CEP dataset is significantly lower than the prediction error of SINet on the two other relatively smaller datasets. This also justifies the use of large dataset as the source dataset while using transfer learning to a smaller target dataset.

4.3.3. Impact of Transfer Learning

We also investigated the impact of transfer learning from the DFT-computed dataset of Harvard CEP to the relatively smaller DFT-computed and experimental datasets from HOPV. For the experimental data having only 243 samples, the MAPE in case of SINet decreased significantly from 2.782% to 1.513% which is around half. We observed similar changes when using just one input or their simple combination as shown in Figure 4.3(a). For the target dataset of DFT-computed dataset with 344 samples, the error for SINet decreased from 2.118% to 1.478% (in Figure 4.3(b)). Such a significant drop in the MAPE for both our target datasets illustrates the efficacy of using transfer learning from large datasets when doing predictive modeling on smaller datasets with a lesser number of samples. The experiments exhibit that when a model is trained on a large dataset (model parameters being initialized from a model trained on large source dataset), it already captures the required features from the dataset which makes it easy to learn the features present in target data from a similar domain on which it is fine-tuned.

4.4. Summarization

In this chapter, we presented a novel approach of predictive modeling for HOMO values of donor molecules for the generation of OPV candidates by leveraging both large DFT-computed dataset and relatively smaller DFT-computed and experimental datasets using both types of input representation- SMILES and InChI, using the concept of transfer learning with deep neural networks. For the source dataset, we leveraged the Harvard CEP dataset which contains millions of OPV candidates with the DFT-computed HOMO values. For the target dataset, we used the DFT-computed and experimental data from HOPV which contains relatively smaller data- 344 and 243, respectively.

Our results demonstrate significant benefit from the use of both types of input representations as well as from transfer learning from a larger dataset. It showcases that leveraging machine learning with computational and experimental chemistry can play an essential role in the expedition of a systematic design of high-efficiency OPV materials, and holds significant promise as a potential solution to future energy needs. The search process for the donor cells with high HOMO values can be made faster by leveraging transfer learning from a larger calculated dataset to a small well-curated experiment-theory calibrated dataset, and this exposes an exciting area in materials discovery, and in particular for solar cell technology. Further, as our approach is based on simple text representations, it is easier for chemists to explore adding or removing subgroups to the chemical compounds to explore the impact on power efficiency instead of performing elaborate experiments.

CHAPTER 5

Development of machine learning-based surrogate model for additive manufacturing simulations

5.1. 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 [138]. 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) [139] 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) [140, 141] 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 [142, 143]. 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) [144, 145], 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 in AM processes which in turn can predict presence of residual stress [146]. Residual stress caused in AM is the critical issue for fabricated metal parts since steep residual stress gradients generate distortion which dramatically deteriorate the functionality of the parts.

The proposed approach uses extremely randomized trees (ERTs) [100], a tree-based ensemble algorithm to iteratively train 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 error (% MAE) below 1% for predicting temperature profiles for AM processes. The 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://github.com/paularindam/ml-iter-additive`.

5.2. 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 to the application of machine learning in materials informatics.

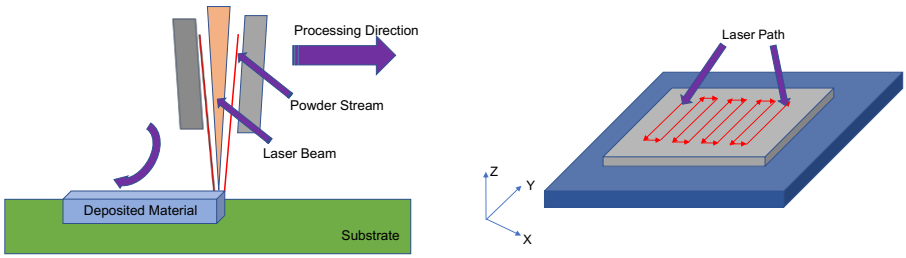
5.2.1. Additive Manufacturing: Overview

The initial development process for creating a three-dimensional object using computer-aided design (CAD) for a layer by layer deposition was realized due to a desire for rapid prototyping [147, 148]. It reduced the time-cycle of realizing an initial prototype after the conception of design by engineers [149]. 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 [150, 151]. 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 [152]. Besides, AM processes can significantly reduce the total amount of tooling required and its impact on the cost [153]. 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 [154].

5.2.2. Direct Metal Deposition

DMD [155] is an additive manufacturing technology using a laser to melt metallic powder. DMD processes can produce fully dense, functional metal parts 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 [156]. 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



(a) DMD overall setup (b) DMD Laser Path on Substrate

Figure 5.1. 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.

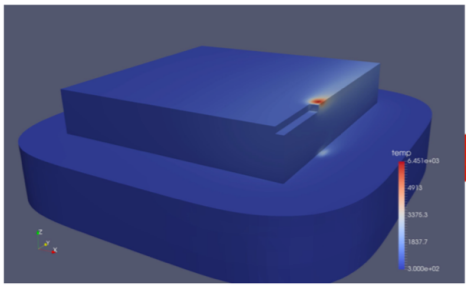


Figure 5.2. The simulated metal surface built using DMD is depicted in the figures. The first figure demonstrates the metal created using DMD on the substrate with the temperature scale. The color of the metal surface indicates the spatio-temporal characteristic of the DMD process.

to form a raised portion of the material. The nozzle is moved over the substrate using a computer-controlled positioning system to create the desired geometry. This is illustrated in Figures 5.1 and 5.2 that depict the DMD process and laser motion, and the metal surface built across layers, respectively.

5.2.3. 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 [157]. 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 a FEM framework that solves transient heat transfer equations for metal powder-based AM processes such as Directed Energy Deposition (DED) [158] and Selective Laser Melting (SLM) [21]. Although an accurate thermal analysis of AM provides vital information for determining microstructure evolution and mechanical performance of the part [141, 159], this kind of analysis can take weeks or months of computing time and therefore too computationally expensive for large-scale problems or optimization purposes [160]. 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.

5.2.4. 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 [23]. Exploring and harnessing the association between processing, structure, properties, and performance is a critical aspect of new materials exploration [19, 97, 161, 162]. Data-driven techniques provide faster methods to know the important properties of materials and to predict feasibility to synthesize materials experimentally. This can expedite the research

process for new materials development. Many initiatives to computationally assist materials discovery using ML techniques have been undertaken [88, 94, 95, 103, 104, 163–170].

There has been some limited work on the application of ML techniques for AM processes. Mozaffar et al. [171] proposed a data-driven approach to predict the thermal behavior in a directed energy deposition process for various geometries using recurrent neural networks. The proposed approach mapped the position of a point on the printing surface, the time of deposition, the distance of the closest cooling surface, and laser parameters with the thermal output. Baturynska et al. [172] propounded a conceptual framework for combining FEM and ML methods for optimization of process parameters for powder bed fusion AM. Choy et al. [173] designed a novel recurrent neural network architecture 3D recurrent reconstruction neural network (3D-R2N2) that learned mapping from images of objects to their underlying shapes in an AM simulation environment. Scime et al. [174–176] developed supervised as well as unsupervised models for detecting irregularities and flaws on the laser bed during the AM process.

5.3. 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.

5.3.1. 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 the edge length of 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.

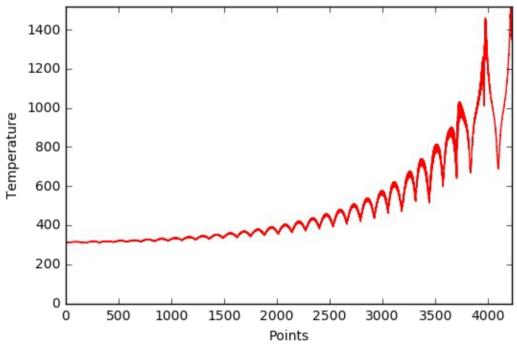
The voxel edge length of 0.5 mm chosen in this work is fairly coarse. However, the time taken for a simulation exponentially increases as the mesh voxel size is made finer. For instance, if we reduce the mesh size to half or 0.25 mm, the FEM simulation would take 9 hours. Moreover, the number of data points is in the order of $\mathrm{O}(n^2)$ 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 timestep, 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 $\mathrm{O}(n^2)$. 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 timestep does not create the same number of voxels as the simulation mimics the weaving (zigzag) motion of the laser (illustrated in Figure 5.1(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 train and evaluate the proposed approach rigorously.

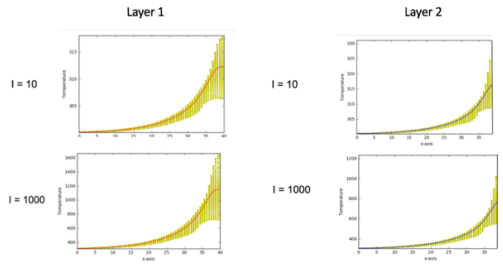
5.3.2. Data Pre-processing

Figure 5.3(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 the 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 5.3(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 the 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 change 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



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



(b) Temperature Profile across different laser intensities and layers

Figure 5.3. Temperature profiles for the DMD process. The temperatures are in Kelvin (K) scale.

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 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. Figure 5.4 depicts the cross-section of the AM-surface to represent conduction of heat between neighboring voxels.

5.3.3. Voxel Categories

We classify the voxels into five 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 five categories.

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

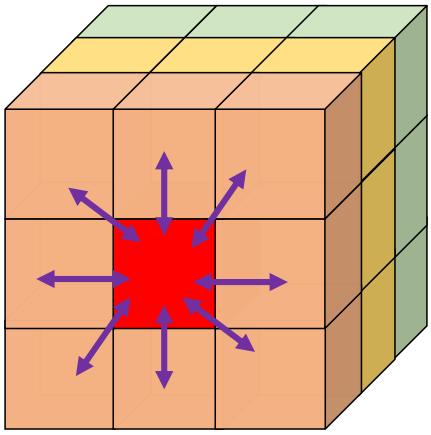


Figure 5.4. Illustration of the cross-section of the AM-surface to represent conduction of heat on target voxel (labeled in red) from neighboring voxels. However, as this is a 2D cross-section of a voxel, there are eight neighboring voxels indicated by arrowheads. In three dimensions, a voxel is surrounded by 26 neighboring voxels. 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 (Vertical): neighboring voxel on the z-axis missing
- 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 an 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

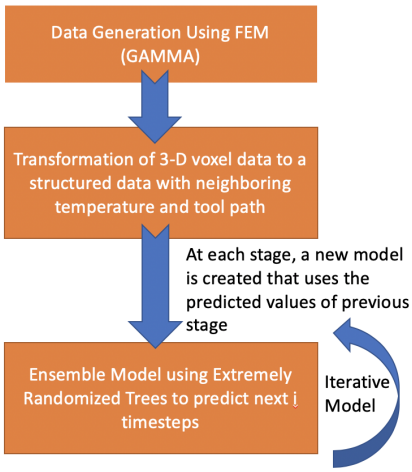


Figure 5.5. The overall methodology of the proposed multi-stage iterative model for predicting temperature profile of an additive process

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 specific 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.

5.4. Method

The motivation and methodology of the proposed iterative approach are outlined in this section. Figure 5.5 illustrates the flow diagram of the proposed methodology.

5.4.1. Motivation for real-time system

Control systems in manufacturing can be divided into two broad categories [177]. The first class is error-based control systems in which changing parameters (parameters of manufacturing machine such as laser power, speed) are estimated and based on the error values from the experiment, the initial guess is corrected until the desired criteria is met.

The second class is model-based in which instead of estimating the initial value of machine parameters, they will be determined by a model.

While an error based control system can be useful in many applications such as motion control, its application in AM process parameter control is not common because a significant deviation 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 where the model is developed by training a machine learning algorithm.

5.4.2. 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 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.

Table 5.1. Comparison of performance for different machine learning algorithms with corresponding R^2 and % 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

Further, algorithms based on autoregression and moving average such as ARIMA [178] would not be able to capture spatial non-temporal relationships. This is also evident from our benchmarking experiment in Table 5.1. We considered two metrics R^2 (coefficient of determination) and % MAE to evaluate the performance of the models.

Algorithms using an ensemble of decision trees have achieved state of the art results for various machine learning tasks [179]. 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 [180, 181]. While AdaBoost and XGBoost are tree-based ensemble boosting algorithms 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 the form of trees with limited depth. Boosting models are sequential learners and harnesses weak learners in sequence. As bagged models use many weak tree-based learners in parallel, and hence can be parallelized in the order of the number of processors. As the 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 5.1 demonstrates the performance of all the different algorithms trained on the 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 can utilize 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 autoregressive moving average modeling [182] 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 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 5.6 demonstrates the iterative learning process of our proposed model.

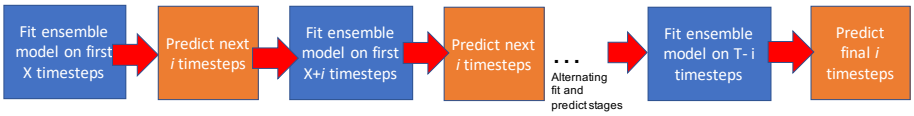


Figure 5.6. The proposed model using ERTs to predict temperature profiles for additive manufacturing processes. It is to be noted that the number of data-points predicted at each step is not the same as the number of data-points for each voxel. This is because the model predicts not only the temperature of the newly created voxels but also the temperature of the same voxels present in the training set at a later time-step.

5.5. Experiments & 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 librarywas used for processing and converting the voxel data. The data preprocessing, as well as most of the regression models, were implemented using Scikit-Learn [64]. The XGBoost package [123] was utilized for creating the xgboost model. The ARIMA model was trained from the statsmodels package [183].

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 5.2,

Table 5.2. Comparison of combinations of time-steps used for training and test in the iterative model (with corresponding R^2 and % MAE). 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	1***% 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 5.3. 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 % MAE) 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.

1***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

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 5.3 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

Table 5.4. 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 5.5. 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

instance, the first 200 timesteps would represent a few completed layers and an incomplete layer. The same intuition follows for the timesteps in the test set. By training on different timesteps allows us to generalize the framework to different shapes. Although the direct model is much faster, the iterative model performs much better than the direct model. For instance, 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 %.

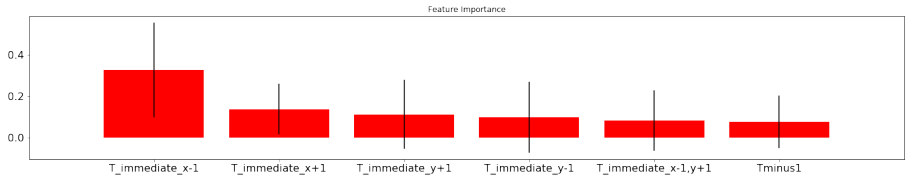
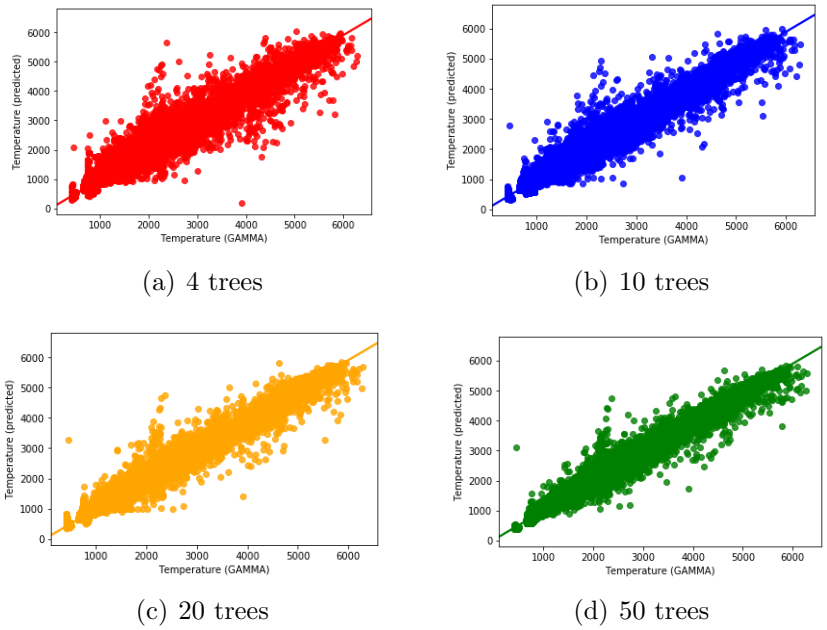


Figure 5.7. The feature importance for the top input features in the ensemble iterative approach



(a) 4 trees (b) 10 trees

(c) 20 trees (d) 50 trees

Figure 5.8. Scatterplot for predicted vs. FEM temperatures. As the number of estimators/trees increase, the prediction accuracy improves.

The results in Table 5.4 illustrates 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 5.5 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 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 5.7 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 5.2b) and is therefore agnostic of the directionality in the y-axis. Figure 5.8 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 a high bias on their own but have low bias overall as an ensemble.

The primary motivation of this work was to develop an 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, but they are also 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.

5.6. Summarization

This chapter 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.

CHAPTER 6

Microstructure Optimization with Constrained Design Objectives using Data-Driven Sampling

6.1. Introduction

One of the primary aims of materials science and engineering research is to understand the association between materials' processing, structure, properties, and performance [4, 7, 8, 11, 19, 161, 184–187]. It is recognized that even for a particular alloy system, variability in microstructure leads to a wide range of materials properties, and it substantially impacts the materials' performance, especially under extreme conditions. Thus, optimization of the microstructure can significantly improve the materials' performance. It is even more pertinent for sensitively engineered components that use magnetostrictive materials. Magnetostrictive materials undergo a change in shape or dimensions in response to a magnetic field. Further, such materials can respond to external stresses by altering their magnetic states. The state of the art of magnetostrictive materials and their applications in a large variety of engineering applications was discussed by Olabi and Grunwald [188]. The authors also showed improvement in material features with the use of magnetostrictive materials. The magnetostrictive properties of different materials such as cubic Laves phases such as TbFe2, Terfenol D, and SmFe2, as well as Fe-X alloys based on Fe-Ga and Fe-Al, were presented by Grossinger et al. [189]. The design of magnetostrictive actuators and transducers has been discussed in literature [188, 190–192];

however, the design of microstructural properties of magnetostrictive materials has not been studied extensively yet. Galfenol is one such example of a magnetostrictive material [1, 193]. It has been widely used in aerospace applications as a sensor material in beam shaped structures. Galfenol can be processed using conventional rolling and wire drawing equipment, and it can be machined using conventional mills and lathes, and welded to a wide array of materials. The potential of Galfenol to develop desired anisotropic properties and flexibility regarding processing makes it a lucrative material. The single crystals of Galfenol material can provide large magnetostriction; however, their preparation is expensive. It is possible to develop comparable polycrystalline textured Galfenol material as expensive single crystals by applying thermomechanical processes such as rolling and extrusion [194–196]. However, control and prediction of the large changes in properties such as magnetostriction and yield strength during thermomechanical processing can be difficult. For instance, warm rolled and annealed specimens retain high magnetostriction but are quite brittle; whereas, cold rolled specimens have high yield strength but lose their magnetostriction [197, 198]. Experimental studies suggest that internal inhomogeneous strains introduced by microstructural changes play a major role in determining the final magnetostriction in Galfenol [199]. The computation of magnetostrictive strain of a polycrystalline Galfenol material was studied before by Kumar and Sundararaghavan [193].

The orientation distribution function (ODF) is used to represent the microstructure. The ODF represents the volume fractions of the crystals of different orientations in the microstructure. The complete range of properties obtainable from the space of ODFs is represented using property closures, approximated by the space defined with either upper or lower bound of a given property [1]. Upper bound closure of stiffness values represents

the range of properties obtainable by the upper bound homogenization relation while a lower bound closure of compliance values shows the properties obtainable by the lower bound homogenization equation. An approach that is gaining popularity in new materials development is selective optimization of certain properties of a material in a particular direction or plane while sacrificing the properties across other directions or planes that are not as important for the design problem [2].

There have been few efforts to optimize the microstructure to satisfy a given set of desired properties. Liu et al. [12] achieved this by directly sampling the ODF space using a data mining methodology. Some researchers have adopted sampling within the property hull and use a Fourier basis for discretizing the ODF [200–202]. In [1, 203], Acar et al. derived an upper bound solution approach starting with generating samples in the space of macro elements (Young's modulus and shear modulus parameters) and then identified multiple optimal solutions through a linear solver. Acar et al. in [2] formulated a Linear Programming (LP) solution based method for constructing property closures (for the homogenization relations considered here) by establishing the smallest convex region enveloping single crystal property points.

However, all these approaches used for constrained microstructure optimization lead to only one or in some cases, a handful of solutions. Further, the process for obtaining multiple solutions is often a trial and error based method. On the other hand, conventional and economical manufacturing processes, such as metal forming and heat treatment can generate only a limited set of microstructures [197, 199, 204]. Moreover, it may not be economically feasible to manufacture a single design solution [161]. Thus, there is a big incentive for developing approaches that can conceive a spectrum of optimal structures.

This work proposes a data sampling based scheme to find numerous near-optimal microstructures to maximize yield stress given vibrational design constraints. The proposed framework involves developing and executing sampling algorithms to generate possible ODF solutions satisfying the process limitations. The sampling algorithms developed in this work, *partition* and *allocation* schemes, are complementary to one another and ensure sampling of the entire feature space. Data points satisfying both the bending and torsional frequency constraints are generated. The proposed data sampling methodology outperforms (or on part with) other optimization techniques and provides 2-3 order of magnitude more near-optimal solutions. Further, our approach opens up additional opportunities for reducing the dimensionality of microstructure space to accelerate the process of achieving solutions that satisfy all the constraints by isolating ODF dimensions that are non-zero across a majority of near-optimal ODF solutions. The solution methodology presents an extensive approach, and thus it can be applied to different ODF representations such as finite element discretization and Fourier series expansion.

6.2. Background

6.2.1. Property representation in Rodrigues' space

The alloy microstructure consists of multiple crystals where each crystal has its distinct orientation. The ODF represents the volume fractions of the crystals of different orientations in the microstructure. The microstructure of the Galfenol alloy system in this work is modeled using ODFs [205–208] which are represented by axis-angle parameterization of the orientation space, as proposed by Rodrigues [209]. Angle-axis representations elucidate an alternate way of representing orientations compared to Euler angles. The

Rodrigues' parameterization is created by scaling the axis of rotation n as $r = ntan\frac{\theta}{2}$, where θ is the rotation angle.

The ODF, a primary concept in texture analysis and anisotropy, is defined based on a parameterization of the crystal lattice rotation. Orientation distributions can be described mathematically in any space appropriate to a continuous description of rotations [205, 206, 209]. The orientation space can be reduced to a subset called the fundamental region, as a consequence of crystal symmetries. Each crystal orientation is depicted uniquely inside the fundamental region by a parameterization coordinate for the rotation r. The ODF, represented by $A(r)$, is the volume density of crystals of orientation r. The fundamental region is discretized into N independent nodes with N_{elem} finite elements and N_{int} integration points per element. A detailed explanation of the ODF discretization and volume averaged equations has been provided in [1, 2, 12, 203]. A single particular orientation or texture component is represented by each point in the orientation distribution. The orientation distribution information can be used to determine the presence of components, volume fractions and predict anisotropic properties of polycrystals. Although the term distribution function is used for ODFs, this is distinct from "distribution function" used for cumulative frequency curve in statistics. The ODF is a probability density but is constrained such that it is normalized to unity over the fundamental region. Figure 6.1 represents the finite element discretization of the orientation space of BCC Galfenol.

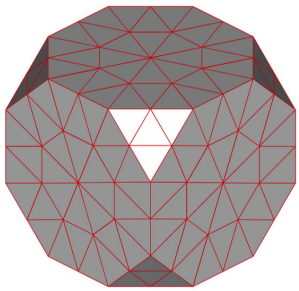


Figure 6.1. Finite element discretization of the orientation space of BCC Galfenol

The microstructure of an alloy is comprised of multiple crystals, and each crystal has an orientation. The generalized Hooke's law for the agglomeration is expressed as:

$$< \sigma_{ij} >= C^{eff}_{ijkl} < \varepsilon_{kl} > \tag{6.1}$$

ε_{kl} and σ_{ij} represent the volume-averaged strain and yield stress of the agglomeration. C^{eff}_{ijkl} represent the tensor for effective stiffness in the given coordinate system. C^{eff} is the average over aggregate of the crystals [2, 210] where $< C >$ represents stiffness tensor for each crystal.

$$C^{eff} =< C > \tag{6.2}$$

The averaging is performed over an aggregate of the crystal in a macro-scale elementary volume. Crystal size and shape are ignored, and homogeneous deformity is assumed. The ODF represents the volume density of each orientation in the microstructure. $\chi(r)$ represents the orientation dependent property for single crystals and $< \chi >$ depicts the

expected value.

$$< \chi >= \int_R \chi(r) A(r,t) dv \tag{6.3}$$

Using this parametrization, any polycrystal property can be expressed in a linear form as follows, where $A(r_m)$ is the value of the ODF at the m^{th} integration point with global coordinate r_m of the n^{th} element, —J_n— is the Jacobian determinant of the nth element, w_m is the integration weight associated with the m^{th} integration point, and $\frac{1}{(1+r_m.r_m)^2}$ represents the metric of Rodrigues parameterization.

$$< \chi >= \chi(r) A(r,t) dv = \sum_{n=1}^{N_{elem}} \sum_{m=1}^{N_{int}} \int_R \chi(r_m) A(r_m) w_m |J_n| \frac{1}{(1 + r_m.r_m)^2} \tag{6.4}$$

A (which symbolizes the ODF) is a function of orientation r and time t during processing that satisfies the following normalization constraint :

$$\int_R A(r,t) dv = 1 \tag{6.5}$$

The complete range of properties obtainable from the space of ODFs is represented using property closures, approximated by the space between upper and lower bound of the given property [1].

$$< C >= \int_R CA dv \tag{6.6}$$

The upper bound homogenization relation (above) is based on the assumption of constant strain throughout the thickness of the beam and is represented by the upper bound closure of stiffness values. The upper bound average or the Voigt average [211] is calculated by averaging the particular property (in this case, stiffness) by multiplying the ODF vector with the property vector. However, the lower bound approach (below) is based on the assumption of constant stress throughout the plate thickness. For the lower bound average or the Reuss average [211], the inverse of the given property is averaged. For instance, in the equation below, compliance (C^{-1} or S), the inverse of stiffness, is averaged by using the lower bound approach and the equation is written for the compliance matrix. $< C >$ and $< C^{-1} >$ represent the volume-averaged macroscopic stiffness formulation in C and C^{-1} space. C^{-1} refers to compliance.

$$< C^{-1} >= \int_R C^{-1} A dv \tag{6.7}$$

The yield stress is computed for the upper and lower bound approaches in terms of single crystal yield strengths along the beam axis as follows:

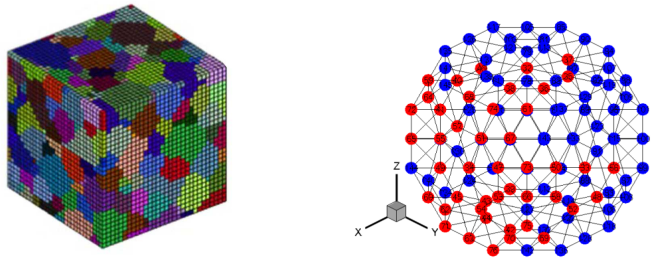
$$< \sigma_y >= \int \sigma_y A dv \tag{6.8}$$

$$< \sigma_y^{-1} >= \int \sigma_y^{-1} A dv \tag{6.9}$$

6.2.2. BCC Galfenol

Galfenol is a general name for an iron-gallium alloy, and the name was first associated in 1998 when it was discovered that adding gallium to iron increases its magnetostrictive effect [212, 213]. A magnetostrictive material is used to harvest vibrational energy because of its property to change shape in response to a magnetic field. Galfenol also responds to external stresses by altering its magnetic state [214]. Researchers have found Galfenol to demonstrate magnetostrictive strains of up to 400 ppm in single crystal form (which is more than ten times that of α-Fe [12]). Moreover, processing Galfenol does not need any customized equipment. It can be processed using conventional rolling and wire drawing equipment, and it can be machined using standard mills and lathes, and can also be welded to a variety of materials [215]. Galfenol converts applied mechanical energy with high efficiency (around 70 percent) into magnetic energy and vice versa [216]. Researchers have found that groups of contorted cells respond to a magnetic field by rotating their magnetic moments to align with the field which in turn, changes the exterior dimensions of the crystal. This contortion from the α-Fe structure is responsible for Galfenol's superior performance [217]. Adding gallium generates imperfections in iron's otherwise orderly lattice thus improving the magnetostrictive property of the resultant alloy [218]. Single crystals of Galfenol impart large magnetostriction, but the preparation of monocrystal Galfenol is expensive. Hence, there is an impetus for the development of polycrystalline Galfenol with favorable properties for various design problems [197, 199, 204]. Figure 7.2 (a) represents the polycrystalline microstructure of Galfenol, with different colors representing different crystal orientations. For this BCC structure, the Rodrigues fundamental region includes 76 independent nodal points (ODF values) as shown in Fig. 7.2 (b). It

is to be noted that the red nodes in Fig. 7.2 (b) are indicating the 76 independent ODF values, and the ODF values of the blue nodes can be computed using the crystallographic symmetries.



(a) BCC Galfenol microstructure (b) ODF representation indicating the location of 76 independent nodes in the orientation space in red

Figure 6.2. Finite element discretization of the orientation space of BCC Galfenol

6.3. Problem Statement

We aim to explore the microstructure design constraint of a cantilevered Galfenol beam for a vibration tuning problem with yielding objective. The vibration tuning puts a restriction on the ODF solutions to have a finite number of directions in the solution space. The number of independent ODF values is 76 at this time since Galfenol has a BCC structure [12]. The design objective is determined as the maximization of yield stress while the first bending and torsional frequencies are constrained for vibration tuning. The

primary goal of the problem is to find the best microstructure design that maximizes the yield stress of the beam and satisfies the given vibration constraints.

The rationale behind constraining the operating frequencies is to eliminate possible dynamic instabilities, for instance, in sensor materials in aircraft beams [219, 220]. The main goal of the problem is to find the best microstructure design that maximizes the yield stress of the beam and satisfies the given vibration constraints.

The torsional and bending frequency constraints are given by the following equations:

$$\omega_{1t} = \sqrt{\frac{G_{12}J}{\rho I_p}} \tag{6.10a}$$

$$\omega_{1b} = (\alpha L)^2 \sqrt{\frac{E_1 I_1}{mL^4}} \tag{6.10b}$$

$$\text{where } \alpha L = 1.87510 \tag{6.11}$$

Here $G_{12} = 1/S_{66}$, $E_1 = 1/S_{11}$, and S being the compliance elements ($S = C^{-1}$), E_1 being the Young's modulus along axis-1 and G_{12} being the shear modulus in 1-2 plane. In these formulations, J is torsion constant, ρ is density, I_p is polar inertia moment, m is unit mass, L is length of the beam and I_1 is moment of inertia along axis-1. The mathematical formulation of the optimization problem is given below:

$$\max \ \ \sigma_y \tag{6.12}$$

$$A \geq 0 \tag{6.13}$$

$$\int A dv = 1 \tag{6.14}$$

The optimization problem includes the unit volume constraint by definition (Equation 6.14). The other constraints are the first natural frequencies to tune the beam vibration. In this problem, the length of the beam is taken as L=0.45 m, and the beam is considered to have rectangular cross section with dimensions a=20 mm and b=3 mm. The values of stiffness parameters for Galfenol single crystals are taken as $C_{11} = 213$ GPa; $C_{12} = 174$ GPa and $C_{44} = 120$ GPa [1, 2, 203]. C_{11}, C_{12}, C_{13} and C_{14} comprise the most dominant elements in stiffness matrix, a measure of the durability of a given material. The stiffness values of the polycrystal are computed using the upper bound averaging (C-space) while the lower bound (C^{-1}-space) computation provides the compliance parameters. Figure 7.1 depicts the geometric representation of Galfenol beam vibration problem. There are two sets of constraints presented below. Each set of constraint has a lower and upper bound on the torsional and bending frequencies.

First set of constraints:

$$\textit{subject to } 19.5 \text{ Hz} \leq \omega_{1t} \leq 21.5 \text{ Hz} \tag{6.15a}$$

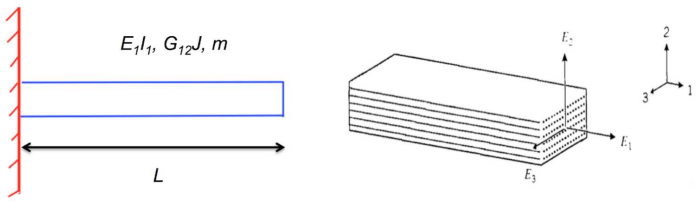


Figure 6.3. Geometric representation of Galfenol beam vibration problem

(6.15b) *subject to* 120 Hz $\leq \omega_{1b} \leq$ 122.5 Hz

Second set of constraints:

(6.16a) *subject to* 21.5 Hz $\leq \omega_{1t} \leq$ 23.5 Hz

(6.16b) *subject to* 100 Hz $\leq \omega_{1b} \leq$ 114 Hz

It is important to note that both sets of constraints are specimens, and factual constraints may differ based on the real material design. Nonetheless, they are representative of a real-world design problem for magnetostrictive materials where there are bounds on first natural frequencies.

6.4. Method

An overview of the proposed system is first presented and then the algorithms proposed for sampling the ODF space for the given problem are explained.

6.4.1. Overview of the System

We propose a two-step data-driven solution scheme to find optimal microstructure satisfying performance requirements, and design and manufacturing constraints. The first phase of the approach involves developing and executing sampling algorithms to generate possible ODF solutions meeting the process limitations. The sampling algorithms i.e. *partition* and *allocation* scheme complement one another and ensure sampling the entire feature space. *Partition* warrants that different permutations of non-zero ODF dimensions are explored for a given set of ODF dimension. *Allocation* guarantees that all the ODF dimensions are explored sufficiently.

In the second step, data points are generated by satisfying both the bending and torsional frequency constraints. A future direction for reducing the dimensionality of microstructure space is highlighted that can accelerate the process of achieving solutions satisfying all the constraints by isolating ODF dimensions that are mostly non-zero across a majority of near-optimal ODF solutions. Figure 6.4 illustrates the steps in the proposed framework in a flow-diagram.

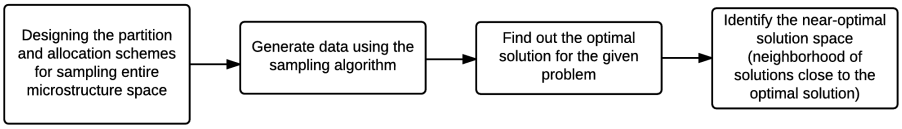


Figure 6.4. Flow diagram of the proposed methodology. Upper and lower bound approaches for both sets of constraints are repeated for both problems.

6.4.2. Algorithms

Two sampling techniques *partition* and *allocation* algorithms developed in the proposed work are presented. The algorithms ensure to address the problem of sufficiently sampling the problem space and generate ODFs fulfilling the constraints in the problem objective.

6.4.2.1. Partition Method. In this method, k small segments that add up to 1, where k can vary between 1 to D where D is the number of dimensions. For HCP Titanium structure, D is 50 for coarse mesh and 388 for finer mesh. We consider the unit length 1 divided into k random intervals or making k random cuts between the interval [0,1], where k is the dimension of ODF. It is iterated from 1 to D-1 with an increasing number of samples generated with regards to k and then downsampled to 1000 for each iteration, except when k=1, D samples exist and are all used.

Figure 6.5. Partition Algorithm : The unit length is divided into k small segments. k-1 random numbers are used as split points to partition unit length.

6.4.2.2. Allocation Method. This randomly generates k values at a time, where k can vary between 1 to D where D is the number of dimensions. In this algorithm, k is the intended maximum number of dimensions for the ODF. The sum of the product of the volume fraction (vf) and density functions (df) across each dimension must add up to 1. Therefore, we continue selecting a value until k values are selected, or the remainder is sufficiently small. k=1 is the trivial case in which the product of the vf and df equal to 1.

Algorithm 1 Partition algorithm

```
0: procedure PARTITION
0:   D ∈ Z
0:   for k ∈ {1, ..., D − 1} do
0:     for i ∈ {1, ..., 1000} do
0:       for cut ∈ {1, ..., k − 1} do
0:         Make an arbitrary cut
0:       end for{Sample with k cuts generated}
0:     end for{1000 samples with different cuts}
0:   end for
0:   return
0: end procedure=0
```

Both the partition and allocation methods are based on the heuristic that in a valid microstructure obeying all the constraints, only a few of all the dimensions of the ODF vector is non-zero. However, these two methods are complementary or reciprocal to each other and ensure that the entire feature space is sampled sufficiently. While the allocation method attempts to find a minimal subset of ODF dimensions that would be non-zero

Algorithm 2 Allocation algorithm

```
0: procedure ALLOCATION
0:   Generate a random k ∈ {1, ..., 76}
0:   Sum ← 0
0:   for i ∈ {1, ..., k} do
0:     Sum ← Sum + vf(i) * df(i)
0:     remainder ← 1 − Sum
0:     if remainder<ϵ then {ϵ:very small value}
0:       break
0:     else
0:       continue
0:     end if
0:   end for
0:   return
0: end procedure=0
```

generating a polycrystal solution, the partition method seeks to widen the search across all the 76 dimensions.

6.5. Results

In this section, we evaluate the proposed data-driven approach in yielding optimal and near-optimal solutions and find that it outperforms or matches previous state-of-the-art methods and produces numerous near-optimal solutions which is one of the most significant contributions of this study. Table 7.1 presents the total number of near-optimal solutions, or in other words, solutions that are proximal to the optimal solutions. The near-optimal solutions of this problem correspond to different designs having same or similar values for yield stress. The algorithms were executed to produce around 5 million valid (which obey all the constraints) solutions. It took an average of 112.21 ms and 303.45 ms for generating a valid sample for the partition and the allocation scheme respectively.

Table 6.1. Number of solutions within 0.01%, 0.1% and 0.5% of the optimal solutions. For each set of constraints (Equations 6.15a, 6.16a), 5 million valid data points were generated.

Constraint	**Bound**	**within 0.01%**	**within 0.1%**	**within 0.5%**
1	Upper	3	89	147
1	Lower	9	92	222
2	Upper	7	402	2015
2	Lower	3	116	1579

Optimization techniques including the methods used by Acar et al. such as a genetic algorithm [1] or linear programming [2] based scheme lead to a unique microstructural solution or sometimes a few. Acar et al. found multiple solutions by augmenting the original solution with null space [2]. Acar and Sundararaghavan [2] previously studied an LP approach to identify the optimal processing routes, which can produce the optimum

microstructure designs of the same Galfenol vibration tuning problem. One of the limitations of their approach for vetting equivalent solutions is that it only searches for identical optimal value. However, for practical design applications, a near-optimal solution is adequate as long as the constraints are strictly obeyed, and the near-optimal solutions are proximal to the optimal solution. For all the four problems (upper and lower bound approaches for two sets of constraints), 3-9 near-optimal solutions with a neighborhood of 10^{-4} (from the optimal solution) are discovered. Further, between 89-402 solutions in a neighborhood of 10^{-3} and between 147-1579 solutions in a neighborhood of $5*10^{-3}$, across all the categories are identified. As described before, obtaining multiple optimal solutions are critical as traditional low-cost manufacturing processes can only generate a limited set of microstructures. While a single solution may not be economically feasible to manufacture, hundreds or thousands of near-optimal solutions can accelerate the speed of materials development. Therefore, it provides flexibility to produce solutions which are cost-effective selectively, and improve the overall efficiency of manufacturing immensely.

Liu et al.'s [12, 221] approach of using guided and generalized pattern search methods was compared with the proposed data-driven methodology for the current design problem. However, neither of these approaches converged to an optimal solution for the current problem. Although both problems have a yielding objective for a cantilevered Galfenol beam, the current problem is more convoluted compared to Liu et al.'s because of additional constraints on the first natural frequencies. Pattern search finds a sequence of points to approach an optimal point. Due to the added constraints in the current problem, pattern search algorithms failed to converge to an optimal solution [222]. For pattern search to successfully reach an optimal solution, it requires a series of valid points

at each iteration of the optimization process. Table 6.2 illustrates the optimal yield stress values and Young's modulus, shear modulus, bending and torsional frequencies obtained by the proposed method for both sets of constraints and bounds.

Table 6.2. Summary of the results: The yield stress σ_y, Young's modulus E_1, shear modulus G_{12}, bending ω_{1b} and torsional ω_{1t} frequencies of the optimal solutions generated by the proposed method for both sets of constraints (Equations 6.15a, 6.16a)

Constraint	**Bound**	$\boldsymbol{\sigma}_y$(in MPa)	$\boldsymbol{E}_1$(in GPa)	$\boldsymbol{G}_{12}$(in GPa)	$\boldsymbol{\omega}_{1b}$(in Hz)	$\boldsymbol{\omega}_{1t}$(in Hz)
1	Upper	385.237	209.546	77.313	120.006	21.341
1	Lower	385.113	235.905	82.316	121.272	21.344
2	Upper	388.089	153.160	93.723	102.649	23.497
2	Lower	387.134	184.679	92.772	112.661	23.377

.

In their previous works [1, 203], Acar et al. used a genetic algorithm based scheme to solve the upper bound approach. In a later work, Acar et al. [2] converted the upper bound approach to a lower bound approach that involved converting the problem from stiffness domain to compliance (reciprocal of stiffness) domain. Hence, by converting the original problem into a linear problem, Acar et al. arrived at an LP solution for the constrained microstructure design problem. The proposed data-driven approach is compared with the methods advanced by Acar et al. as their approach outperformed other optimization methods. For the upper and lower bound approaches, our solutions are compared against the genetic algorithm based scheme and LP-based methods respectively. The proposed data sampling approach based on the sampling algorithms surpassed the yield stresses obtained from genetic algorithm based solver for the upper bound approach (as shown in Table 6.3). In particular, we get an improvement of more than 25% for upper bound approach on the first set of objectives against the previous state-of-the art approach. Additionally, the results for the lower bound are comparable to the optimal

values achieved by the LP method (Table 6.4). It is important to note that only the LP solution (used for the lower bound approach by Acar et al. [2]) yields the theoretical maximum value in contrast to the genetic algorithm solver scheme used by them for the upper bound approach [1].

Table 6.3. Comparison of the maximum yield stress achieved for the 2 sets of constraints with the proposed approach and the previous state-of-the art genetic algorithm solver (GA) [1] approach for microstructure design with process constraints(upper bound). The yield stress σ_y, bending ω_{1b} and torsional ω_{1t} frequencies of the optimal solutions generated by both methods. The units for yield stress σ_y is MPa and the frequencies is Hz.

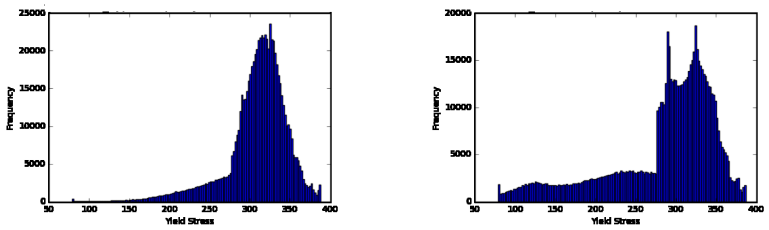
Constraint	**Bound**	$\boldsymbol{\sigma}_y$(current)	$\boldsymbol{\sigma}_y$(GA)	$\boldsymbol{\omega}_{1b}$(current)	$\boldsymbol{\omega}_{1b}$(GA)	$\boldsymbol{\omega}_{1t}$(current)	$\boldsymbol{\omega}_{1t}$(GA)
1	Upper	385.237	384.126	120.006	120.210	21.341	21.498
2	Upper	388.089	308.446	102.589	113.918	23.482	23.485

Table 6.4. Comparison of the maximum yield stress achieved for the 2 sets of constraints with the proposed approach and the previous state-of-the-art LP [2] approach for the microstructure design with process constraints (lower bound). The yield stresses σ_y, bending ω_{1b} and torsional ω_{1t} frequencies of the optimal solutions generated by both methods. The units for the yield stress σ_y is MPa and the frequencies is Hz.

Constraint	**Bound**	$\boldsymbol{\sigma}_y$(current)	$\boldsymbol{\sigma}_y$(LP)	$\boldsymbol{\omega}_{1b}$(current)	$\boldsymbol{\omega}_{1b}$(LP)	$\boldsymbol{\omega}_{1t}$(current)	$\boldsymbol{\omega}_{1t}$(LP)
1	Lower	385.113	385.650	121.272	120.020	21.344	21.500
2	Lower	387.134	387.259	106.519	100.000	23.477	23.499

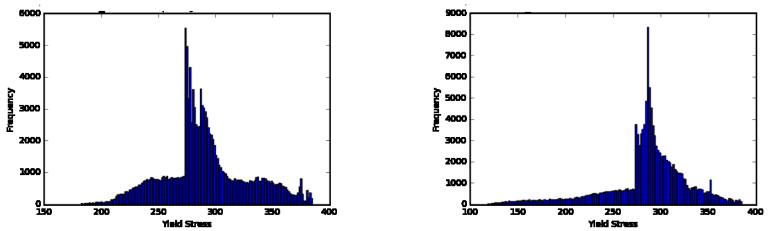
Figures 6.6 and 6.7 represent the frequency distribution of yield stress values for upper and lower bounds for the first and second set of constraints respectively. Figures 6.8 and 6.9 depict the optimal upper and lower bound ODF solutions for the two constraints respectively.

A sensitivity analysis is performed by representing the distribution ODF and frequency plot (inset) of the top/highest 1% yield stress values across the 76 ODF dimensions (Figures 6.10 and 6.11). The figures exhibit the fraction (or percentage) of non-zero ODFs



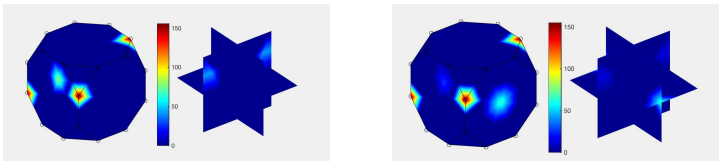
(a) upper bound approach (b) lower bound approach

Figure 6.6. Frequency distribution of yield stress values for first set of constraints



(a) upper bound approach (b) lower bound approach

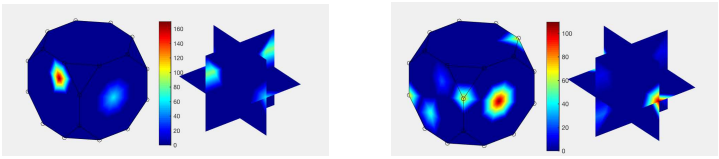
Figure 6.7. Frequency distribution of yield stress values for second set of constraints



(a) upper bound approach (b) lower bound approach

Figure 6.8. Finite element microstructure of optimal ODF examples for the first set of frequency constraints

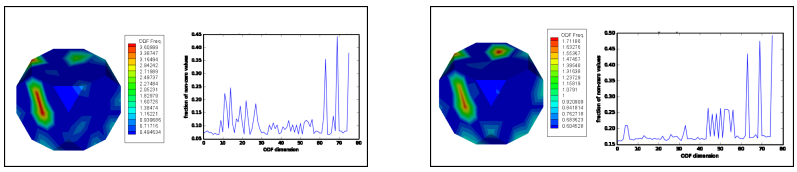
in the ODF vectors that yield high-stress values, in the case of both upper and lower bound solutions for both sets of constraints. The peaks in the frequency plots represent the ODF dimensions that are non-zero across the majority of ODF vectors yielding the highest objective value (in this case, yield stress). The distribution ODFs in these figures



(a) upper bound approach (b) lower bound approach

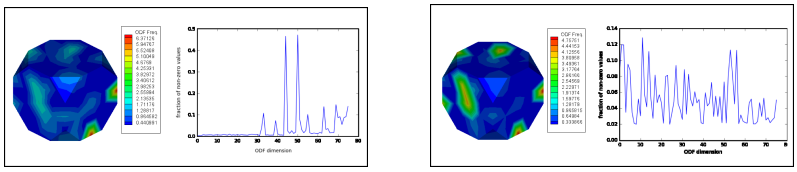
Figure 6.9. Finite element microstructure of optimal ODF examples for the second set of frequency constraints

does not exhibit the actual values. Rather, they represent the percentage of occurrence of the ODF dimensions in top 1% of the solutions. It is observed that the sensitivity of the near optimum solutions to the ODFs for the lower and upper bound approaches are similar, especially for the first set of constraints. Although the computation of intermediate properties in the case of upper or lower bound solutions is different (stiffness and compliance respectively), this is admissible as the same objective function is being solved. The figures signify that a small number of ODF dimensions can predominantly influence the solution space proximal to the optimal value. This can motivate the development of future sampling approaches for ODF vectors to iteratively adapt to sample across only few ODF dimensions instead of all to accelerate the data-generation process.



(a) upper bound approach (b) lower bound approach

Figure 6.10. Finite element discretized sensitivity plots for ODF and frequency distribution(inset) of the top/highest 1% yield stress values across ODF dimensions for the first set of constraints.



(a) upper bound approach (b) lower bound approach

Figure 6.11. Finite element discretized sensitivity plots for ODF and frequency distribution(inset) of the top/highest 1% yield stress values across ODF dimensions for the second set of constraints.

One weakness of the proposed data-driven method is its higher time cost compared to LP methods. However, traditional optimization techniques using combinatorial search methods or evolutionary algorithms are also time-consuming. Our framework attempts to search the entire sample space for attaining the optimal or near-optimal solutions. Besides, it should be emphasized that the proposed sampling algorithms are designed to work even for the more difficult problem of nonlinear optimization. Heuristic search using data-driven approaches is beneficial for solving problems in which the objective function has a non-convex relation to its set of constraints. Another major advantage of the proposed sampling scheme is achieving numerous optimal and near-optimal solutions, that can, in turn, reduce the time and effort for the transition between design and processing.

6.6. Summarization

The selection of materials and geometry to maximize or minimize a given property has been a cardinal problem in materials science. The potential of data-driven approaches for solving a constrained microstructure design problem for both upper and lower bound methods is expounded by the proposed strategy. Our approach arrives at a higher (or in few cases, equivalent) optimal value than the previous state-of-the-art methods. The data

generation strategies attempt to explore the entire sample space and generate numerous near-optimal solutions (about 100-1000, i.e. $2-3$ orders of magnitude more than prior methods). Previous approaches including LP techniques lead to a unique or a handful of optimal solutions. Numerous near-optimal solutions give the flexibility to use traditional low-cost manufacturing processes such as forming and heat treatment. These processes can generate only a limited set of microstructures, and frequently manufacturing from a single optimal solution may not be feasible.

Leveraging data-driven techniques can play an essential role in the expedition of a precise design of materials with process constraints. This study has demonstrated the power of carefully designed sampling approaches by identifying multiple near-optimal solutions for a non-linear optimization problem, and is expected to inspire the development of alternative sampling schemes building upon the ones proposed in this work which can reach optimal solutions faster and deliver numerous near-optimal solutions. Further, with parallel computing technologies becoming inexpensive, especially Graphical Processing Units(GPU) computing, distributed implementations of our algorithm can significantly diminish the optimization time.

The analysis for the constrained microstructure optimization problem depicts that certain combinations of ODF dimensions are non-zero more often in the ODF vector of the near-optimal solutions. The proposed work provides a future direction for feedback aware sampling that can iteratively incentivize distinct ODF dimensions that yield ODF vectors with higher objective value, which can be investigated to accelerate the process of attaining optimal or near-optimal solutions.

CHAPTER 7

Microstructure Optimization with Constrained Design Objectives using Machine Learning-based Feedback-aware Data Generation

7.1. Introduction

Exploring and harnessing the association between processing, structure, properties, and performance is a critical aspect of new materials exploration [4, 7, 8, 11, 19, 161, 162, 185]. Variation in microstructure leads to a wide range of materials properties which in turn impacts the performance. The materials performance can be significantly improved by dovetailing the microstructure [12, 184, 186, 223]. Titanium alloys are used for airframe panels, and optimizing the property is necessary for the safety and performance of the aircraft [224–227]. Furthermore, both the cost of the material and machining for Titanium panels are expensive [228, 229]. Due to their high tensile strength to density ratio, high corrosion resistance, and ability to withstand moderately high temperatures without creeping, titanium alloys are used considerably in aircraft applications. It is also a very ductile material that can be worked into many shapes.

One of the major goals of design optimization in scientific applications is the trade-off of properties based on prioritizing one design goal over others [230, 231]. For microstructure optimization, it can involve enhancing properties in one direction while sacrificing

the properties in other directions where they are not as important for the design problem [232]. Techniques that allow tailoring of properties of polycrystalline alloys involves selection of preferred orientations of various crystals constituting the polycrystalline alloy. This work addresses the problem by tailoring crystallite distribution for specific optimization design problems. The orientation distribution function (ODF) is used to quantify the microstructure [161, 206–208] which represents the volume fractions of crystals of different orientations of the microstructure.

In this work, we aim to explore the microstructure optimization of multiple design problems for a Titanium panel. Two different mesh sizes to represent ODFs are explored in this work - 50 and 388. Three different properties: coefficient of expansion α, stiffness coefficient C_{11} and yield stress σ are optimized. Our data sampling-based methodology not only outperforms or is on par with other optimization techniques in terms of the optimal property value but also provides numerous near-optimal solutions, 3-4 orders of magnitude more than previous methods.

7.2. Problem Statement

We aim to explore the microstructure optimization of multiple design problems for a Titanium panel. Two different mesh sizes to represent ODFs are investigated in this work - 50 and 388. Three separate properties: coefficient of thermal expansion α, stiffness coefficient C_{11} and yield stress σ are optimized. There are four different design problems explored, and both the upper and lower bounds are solved. Figure 7.1 illustrates a section of Titanium aircraft panel and corresponding microstructure cross-section.

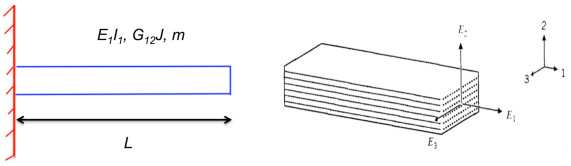


Figure 7.1. Geometric representation of Titanium panel

The ODF values are associated with an orientation of the microstructure. Using the ODF approach is advantageous since the averaged material properties over a microstructural domain can be computed using the homogenization (averaging) equations which are linear with respect to the ODF values. This is true when the effects of crystal size and shape are ignored, and homogenous deformity is assumed in the volume element. Using the homogenization relation, the orientation-dependent averaged material property, $< \chi >$, can be computed using the material property values at different orientations, $\chi(r)$, and the ODF values, A.

$$< \chi >= \int_R \chi(r) A(r) dv,$$

where, the orientation is denoted by r. The ODF representation should satisfy the following volume normalization constraint in the microstructural domain.

$$\int_R A(r) dv = 1$$

The optimization problems of interest aim to identify the best microstructure design to enhance the material properties. Since the ODF values quantify the microstructural texture, the goal is to identify the optimum ODF values for each problem. However, the ODF solution space is high-dimensional, and it leads to an optimization problem with

numerous design variables. Here, one favorable approach would be generating a new solution space, which is called as property closure, which includes the complete range of properties obtainable from the space of the ODFs. In property closure approach, the material properties can be calculated with either upper or lower bound averaging assumption [1]. An example computation of property closure with upper and lower bounds approaches is shown in Fig. 7.3 for stiffness (C_{11}, C_{12} and C_{22}) and compliance (S_{11}, S_{12} and S_{22}) properties. The example computations for the averaged stiffness, $< C >$, and compliance, $< S >=< C^{-1} >$, are given next for the upper and lower bound approaches respectively.

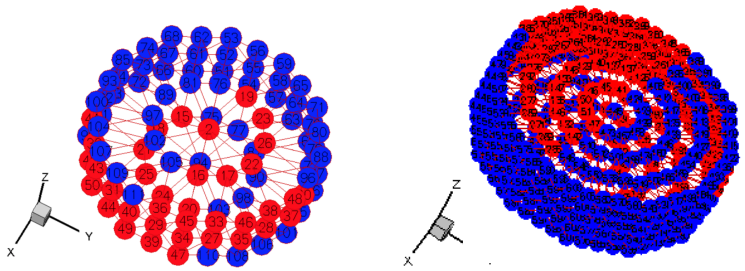
$$< C >= \int_R CAdv$$

$$< S >=< C^{-1} >= \int_R C^{-1}A^{-1}dv$$

$$< S >= \int_R SAdv$$

In the present work, we will utilize both upper and lower bound averaging techniques to identify the optimum microstructure solutions. The material of interest is polycrystalline α-Titanium as shown in Figure 7.2 (a), red color shows independent orientations, blue color shows dependent orientations resulting from the crystallographic symmetries. We will model this hexagonal close-packed (HCP) structure using 111 ODF values defined in the Rodrigues fundamental region as shown in Fig. 7.2 (b). However, we will only use 50 independent ODF values for modeling purpose since the remaining ODF values can be

determined using the crystallographic symmetries. In Fig. 7.2 (b), a finer finite element mesh, that can improve the numerical resolution of microstructural texture representation, having 388 independent ODF values is illustrated.



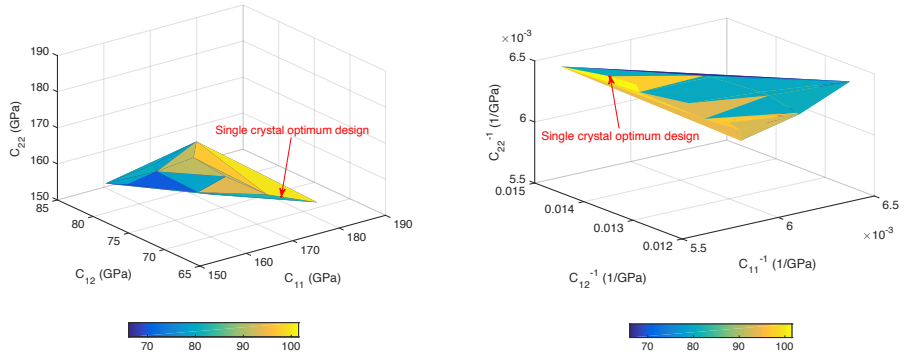
(a) ODF representation indicating the location of 50 independent nodes in the orientation space in red

(b) ODF representation indicating the location of 388 independent nodes in the orientation space in red

Figure 7.2. Finite element discretization of the orientation space of HCP Titanium.

In this work we solve for the best microstructure design that maximizes desired properties which are coefficient of thermal expansion α_x, stiffness coefficient C_{11} and yield stress σ_y and satisfies the design constraints. The material properties of the objective function are computed using the upper bound averaging approach. For design constraints both upper and lower bound averaging approaches are utilized.

Four design problems are presented in this work, and each of the problems are solved using both upper and lower bound approach. Upper bound sub-problems for design problems 1 and 2 are being solved in mesh sizes of both 50 and 388, while the lower bound sub-problems are being solved in 50 dimensions. Both the upper and lower bound design sub-problems 3 and 4 are solved in mesh size of 388. The finer mesh with the 388 ODF values is expected to provide a more accurate representation as the Rodrigues



(a) Property closure for stiffness (C_{11},C_{12} and C_{22}) parameters

(b) Property closure for compliance (S_{11},S_{12} and S_{22}) parameters

Figure 7.3. Property closures in C and S space for HCP Titanium

domain is discretized with more variables. The design constraints of the optimization problems reflect certain stiffness needs of engineering designs.

Problem 1:

$$\textbf{max } \boldsymbol{\alpha}_x$$

Upper Bound: (mesh dimension 50 and 388)

(7.1a) $$\textit{subject to } 161 \leq C_{11} \leq 165 \text{ GPa}$$

(7.1b) $$\textit{subject to } 75 \leq C_{12} \leq 78 \text{ GPa}$$

Lower Bound: (mesh dimension 50)

(7.2a) $$\textit{subject to } 0 \leq C_{11} \leq 125 \text{ GPa}$$

(7.2b) $$\textit{subject to } 90 \leq C_{12} \leq 95 \text{ GPa}$$

Problem 2:

max C_{11}

Upper Bound: (mesh dimension 50 and 388)

(7.3) *subject to* $75 \leq C_{12} \leq 78$ GPa

Lower Bound: (mesh dimension 50)

(7.4) *subject to* $90 \leq C_{12} \leq 95$ GPa

Problem 3: (mesh dimension 388)

max σ_y

Both Bounds:

(7.5a) *subject to* $\leq S_{11} \leq 0.15$ 1/GPa

(7.5b) *subject to* $\leq S_{22} \leq 0.1$ 1/GPa

Problem 4: (mesh dimension 388)

max $\boldsymbol{\sigma}_y$

Upper Bound:

(7.6a) *subject to* $120 \leq C_{11} \leq 130$ GPa

(7.6b) *subject to* $90 \leq C_{12} \leq 95$ GPa

(7.6c) *subject to* $0 \leq S_{11} \leq 0.15$ 1/GPa

(7.6d) *subject to* $0 \leq S_{22} \leq 0.1$ 1/GPa

Lower Bound:

(7.7a) *subject to* $0 \leq C_{11} \leq 125$ GPa

(7.7b) *subject to* $0 \leq C_{12} \leq 75$ GPa

(7.7c) *subject to* $0 \leq S_{11} \leq 0.15$ 1/GPa

(7.7d) *subject to* $0 \leq S_{22} \leq 0.1$ 1/GPa

Apart from the specific set of design constraints for the problem, they should also obey the following generic constraints. It is important to note that the set of constraints for the problems are representative examples, and actual constraints may differ from them in the real design. However, it was ensured that the design constraints resembled real-world problems.

$$A \geq 0$$

$$\int Adv = 1$$

7.3. Method

The proposed methodology is divided into two phases. In the first phase, a data repository is created using two sampling algorithms mentioned in Chapter 6. In the second phase, we evaluate which combinations of ODF dimensions lead to optimal solutions by machine learning. The following flow-diagram illustrates the overall methodology.

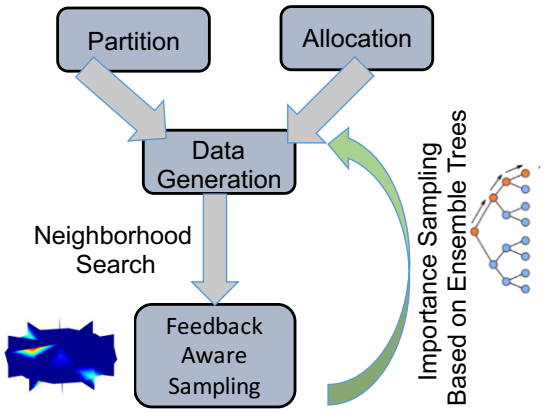


Figure 7.4. Flow diagram of our methodology. The green arrows depict the data generation process, and the orange arrow signifies the feedback-aware sampling.

In the first phase of our methodology, we had generated a dataset using two sampling algorithms. In the second phase, we attempt to investigate which combination of non-zero ODF dimensions lead to optimal or near-optimal solutions. For this purpose, we select the top 10 % and bottom 10 % based on the desired design objective and label them as 'High' and 'Low', and perform random forest-based [233] machine learning models on this data subset, where the ODFs become the feature vector. For instance, in design problem 1, as the objective was maximizing the coefficient of expansion α_x, ODF vectors yielding the highest 10 % and bottom 10 % of α_x are labeled as 'High' and 'Low'. Random Forests are ensemble learning methods that construct multiple decision trees [234] to predict the correct output. The label of the expected output is decided by a vote across the ensemble of decision trees.

The motivation behind this step is to evaluate ODF dimensions which are important for generating optimal solutions. This step extracts the features that are most important for generating 'High' values. However, as the target is to generate a polycrystalline solution, we proceed to the second iteration of sampling. However, during this step, instead of sampling across all ODF dimensions, we select only those dimensions that are advantageous in providing near-optimal solutions.

7.4. Results

In this chapter, we evaluate the proposed data-driven approach for generating optimal and near-optimal solutions. The proposed method is comparable to solutions produced by prior state-of-the-art techniques and delivers numerous optimal or near-optimal solutions with distinct microstructure designs. The near-optimal solutions for this problem

Table 7.1. Number of solutions within 0.01%, 0.02%, 0.05% and 0.1% of the optimal solutions for the fourth set of constraints

		ML-Guided Sampling			
Bound	**Mesh Size**	within 0.01%	within 0.02%	within 0.05%	within 1%
Upper	388	140	280	759	$1.255\text{x}10^{3}$
Lower	388	0	$6.223\text{x}10^{3}$	$1.078\text{x}10^{5}$	$1.084\text{x}10^{5}$

Table 7.2. Comparison of coefficient of expansion α_x, and stiffness parameters (C_{11} and C_{12}) between traditional optimization approaches and ML-Guided Sampling for design problem 1 (Equations 7.1, 7.2)

Bound	**Mesh Size**	**Linear Programming and Genetic Algorithm**			**ML-Guided Sampling**		
		$\boldsymbol{\alpha}_x$(in 1/K)	C_{11}(in GPa)	C_{12}(in GPa)	$\boldsymbol{\alpha}_x$(in 1/K)	C_{11}(in GPa)	C_{12}(in GPa)
Upper	50	$8.5506\text{x}10^{-6}$	161.0000	75.0000	$8.4903\text{x}10^{-6}$	161.0631	75.0450
Upper	388	$8.8560\text{x}10^{-6}$	161.0000	75.0000	$8.8392\text{x}10^{-6}$	161.0519	75.0486
Lower	50	$9.3682\text{x}10^{-6}$	126.6925	90.0000	$9.3790\text{x}10^{-6}$	129.9803	91.6693

Table 7.3. Comparison of stiffness parameters (C_{11} and C_{12}) between traditional optimization approaches and ML-Guided Sampling for design problem 2 (Equations 7.3, 7.4)

Bound	**Mesh Size**	**Linear Programming and Genetic Algorithm**		**ML-Guided Sampling**	
		C_{11}(in GPa)	C_{12}(in GPa)	C_{11}(in GPa)	C_{12}(in GPa)
Upper	50	167.8562	75.0000	167.8538	75.0013
Upper	388	170.2609	75.0000	169.8015	75.0049
Lower	50	144.2199	95.0000	144.1442	94.9546

Table 7.4. Comparison of yield stress (σ_y) and compliance parameters (S_{11} and S_{12}) between traditional optimization approaches and ML-Guided Sampling for design problem 3 (Equation 7.5)

Bound	**Mesh Size**	**Linear Programming and Genetic Algorithm**			**ML-Guided Sampling**		
		$\boldsymbol{\sigma}_y$(in MPa)	S_{22}(in 1/GPa)	S_{12}(in 1/GPa)	$\boldsymbol{\sigma}_y$(in MPa)	S_{22}(in 1/GPa)	S_{12}(in 1/GPa)
Upper	388	423.9396	0.0071	0.0098	423.9396	0.0071	0.0097
Lower	388	423.8462	0.0150	0.1073	422.8327	0.0200	0.0999

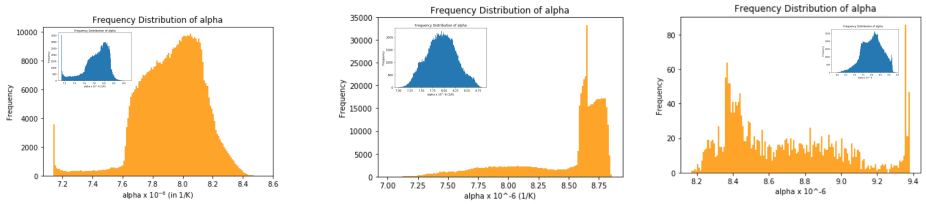
correspond to different microstructure configurations having same or similar values for yield stress. Furthermore, in our study, several different objectives are solved, and the proposed approach is successful for both coarse (50 dimensions) and fine (388 dimensions) meshes for HCP Titanium. Table 7.1 presents the total number of near-optimal solutions, or in other words, solutions that are proximal to the optimal solutions.

Table 7.5. Comparison of yield stress σ_y, stiffness parameters (C_{11}, C_{12}), and compliance parameters (S_{11} and S_{22}) between traditional optimization approaches and ML-Guided Sampling for design problem 4 (Equations 7.6, 7.7)

Bound	**Approach**	**Mesh Size**	**Linear Programming and Genetic Algorithm**				
			$\boldsymbol{\sigma}_y$(in MPa)	C_{11}(in GPa)	C_{12}(in GPa)	S_{11}(in 1/GPa)	S_{22}(in 1/GPa)
Upper	LP	388	421.8096	175.0000	69.6976	0.0075	0.0095
Upper	ML	388	421.8094	174.9997	69.6976	0.0074	0.0094
Lower	GA	388	423.6050	124.8043	78.3030	0.01612	$5.8017\text{x}10^{-8}$
Lower	ML	388	422.8341	119.8148	80.7035	0.0200	0.0999

Acar et al. in their previous works [1, 203] used a genetic algorithm based scheme to solve the upper bound problem. In [2], the upper bound approach was transformed to a lower bound approach by converting the problem from stiffness domain to compliance (reciprocal of stiffness) domain and thereby transforming a non-linear problem into a linear problem that is LP-solvable. In [232], a data-driven approach for arriving at a near-optimal solution was expounded for upper and lower bound problems for optimization of the yield stress of cantilevered Galfenol beam under vibrational constraints. The proposed work improves on the previous methodology by identification of a minimal subset of ODF dimensions using machine learning.

For the upper and lower bound approaches, our solutions are compared against the genetic algorithm based scheme and LP-based methods respectively. The proposed data sampling approach based on the sampling algorithms surpassed the yield stresses obtained from genetic algorithm based solver for the upper bound approach. Additionally, the results for the lower bound are comparable to the optimal values achieved by the LP method. It is important to note that only the LP solution (used for the lower restricted approach by Acar et al. [2]) yields the theoretical maximum value in contrast to the genetic algorithm solver scheme used by them for the upper bound approach [1].



(a) upper bound (mesh size 50)

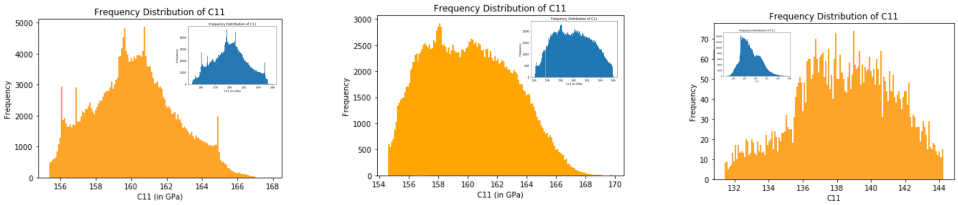
(b) upper bound (mesh size 388)

(c) lower bound

Figure 7.5. Frequency distribution of coefficient of expansion for upper (mesh sizes 50 and 388) and lower bounds (mesh size 50) for first set of constraints (Equations 7.1, 7.2) for ML-Guided sampling. The overall frequency distribution of entire sampling process is presented inset.

Figures 7.5,7.6 represent the frequency distribution for the feedback-driven data-generation of coefficient of expansion and C_{11} for upper (mesh sizes 50 and 388) and lower bounds (mesh size 50) for first set of constraints (Equations 7.1, 7.2) for ML-Guided sampling. Figures 7.7, 7.8 illustrate finite element discretized sensitivity ODF cross-sections (mean and standard deviation) and frequency distribution of the maximal desired values across ODF dimensions for design problem 1 and 2. The frequency distribution and sensitivity plots for design problems 3 and 4 are presented in the Appendix. Examples of finite element microstructure (FEM) cross-sections of near-optimal ODF solutions for all four objective problems are presented in the Appendix.

The potential of our methodology to produce many optimal solutions for the upper bound subproblem in the neighborhood of the LP solution for design problem 1 and 2 for both mesh sizes demonstrate that our method can be advantageous for any mesh size.

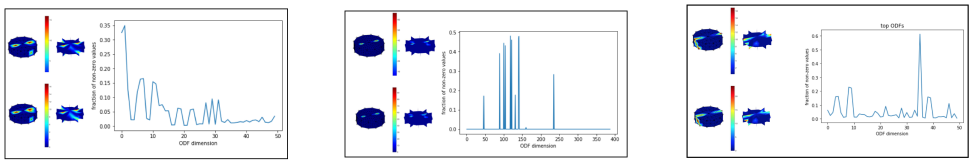


(a) upper bound (mesh size 50)

(b) upper bound (mesh size 388)

(c) lower bound

Figure 7.6. Frequency distribution of C_{11} for upper (mesh sizes 50 and 388) and lower bounds (mesh size 50) for second set of constraints (Equations 7.3, 7.4) for ML-Guided sampling. The overall frequency distribution of entire sampling process is presented inset.

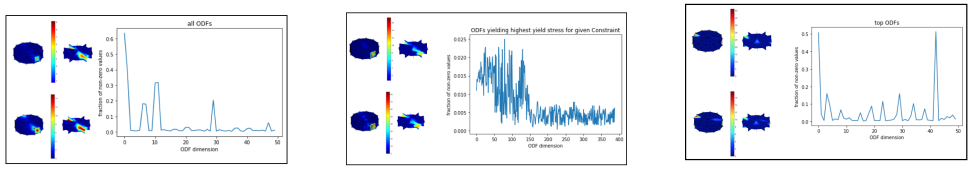


(a) upper bound (mesh size 388)

(b) upper bound (mesh size 50)

(c) lower bound

Figure 7.7. Finite element discretized sensitivity ODF cross-sections (mean and standard deviation) and frequency distribution(inset) of the highest 1% yield stress values across ODF dimensions for design problem 1.



(a) upper bound (mesh size 388)

(b) upper bound (mesh size 50)

(c) lower bound

Figure 7.8. Finite element discretized sensitivity ODF cross-sections (mean and standard deviation) and frequency distribution(inset) of the highest 1% yield stress values across ODF dimensions for design problem 2.

7.5. Summarization

The selection of materials and geometry to optimize desired properties has been a cardinal problem in materials science. The proposed strategy expounds the potential of data-driven approaches for solving a constrained microstructure design objective for both upper and lower bound problems. Our approach is comparable to previous state-of-the-art methods. We outperform the maximum solutions obtained using Genetic algorithms and are close to the theoretical maximum solution obtained using LP. The targeted sampling approach proposed first explores the entire sample space and then selectively generates solutions that optimize the given design objective. The proposed approach generates numerous near-optimal solutions, 3 to 4 orders of magnitude higher than prior methods. Past methodologies including LP techniques lead to a unique or handful of optimal solutions. One of the challenges of inverse materials problems is establishing production feasibility of proposed microstructure design. Many cost-aware manufacturing processes can generate specific microstructures, and thus, selecting from hundreds of thousands of optimal microstructures accelerates the design to deployment step.

CHAPTER 8

Conclusion and Future Work

8.1. Conclusion

This dissertation has presented several works that attempts to address some of the challenges for creating machine learning and data-driven optimization systems for scientific applications. All the works involve developing solutions that accelerate the discovery or prediction tasks in scientific applications. The works represent the wide applicability of data science techniques across domains, disciplines and datasets.

Chapters 2, 3 and 4 presents several techniques for predictive modeling for property prediction tasks. Chapter 2 focuses on development of a generalizable neural architecture for data mining from molecular structures. It proposes a framework of architectures CheMixNet which combines representations using vector-based representation such as fingerprints as well as a text-based representation such as SMILES. We were able to achieve much better results than neural networks that only utilized SMILES or fingerprints. Chapter 3 presents application of extreme random forests for predicting HOMO values of OPV cells from the HOPV dataset. Our models trained on MACCS and Atom-Pair fingerprints outperformed other models trained on neural networks as well as other machine learning algorithms. Further, we studied the correlation of top ranked features from our models with HOMO value, and explained the HOMO values of the molecules in the HOPV dataset. The success of using machine learning models on a small but

well-curated calibrated dataset exposes an exciting area in materials discovery, and in particular for solar cell technology. This, in turn, can provide a path towards solving the world energy problem in a clean and environmentally friendly way. In Chapter 4, we utilize transfer learning to improve our neural network models for predicting HOMO values for the HOPV dataset. We develop another MISO neural architecture SINet which uses two text-based representations SMILES and InChI, and train on the much larger CEP dataset. Thereafter, we use this pre-trained model as a seed model for training on the HOPV datasets. Transfer learning provides significant performance gains on both the experimental and DFT data in the HOPV dataset.

Chapter 5 propounds an iterative ML-based surrogate modeling approach for Additive Manufacturing-simulations. The iterative modeling approach develops an initial model on a small amount of simulation timesteps and predicts the temperature for the next few timesteps. Then, it develops another model that absorbs the predicted values, and further predicts the next few timesteps. We keep repeating this process. This iterative approach outperformed a non-iterative approach, and also achieved mean absolute percentage errors less than 1%.

Chapters 6 and 7 discusses frequency-based and ML-aided approaches to constrained optimization. In Chapter 6, we solve for maximizing yield strength of Galfenol microstructures subject to frequency constraints. Sampling the entire search space exhaustively is difficult. Hence, we develop a targeted sampling approach that extracts dimensions of the ODF that lead to higher objective values, and then sample only across these dimensions. In Chapter 7, we utilize machine learning for narrowing the search space further for several objective functions relating to microstructure optimization in Titanium. The ODF

vector has more than five times the dimension for Galfenol, and hence even a frequency-based approach doesn't suffice. A machine learning-based approach was able to narrow the search space, and sample only on those combinations of ODF dimensions that would lead to optimal values for the objectives.

In the next section, we discuss how our methods can be extended to more general cases and the opportunities for future work.

8.2. Future Work

8.2.1. Chemical Property Prediction

In this thesis, we demonstrate the success of MISO architectures such as CheMixNet (Chapter 2) or SINet(Chapter 4) for predicting chemical properties. One of the future tasks would be develop a method for interpreting the impact of the candidate neural sub-networks on the whole network. For a given prediction, it would be useful to understand which features or which part of the network were more useful. Also, we suggest augmenting ConvGraph and Chemception architectures as candidate input neural networks as part of CheMixNet to generalize not only across fingerprints and SMILES/InChI but also across molecular graphs and images. Further, as chemical significance is present in both the character following as well as preceding a given character in a SMILES string, we believe bidirectional RNNs can perform better than vanilla one-directional RNNs. Lastly, we believe that Hierarchical Attention Networks (HANs) [235] that combine character level and word level sequences for text prediction could present superior performance to the aforementioned architectures.

Directed efforts are needed to standardize the collection and representation of experimental manufacturing and processing data for effective use with machine learning techniques. For the use case of organic solar cells, leveraging machine learning with computational and experimental chemistry could play an essential role in the expedition of systematic design of high-efficiency photo-voltaic materials and holds significant promise as a potential solution to future energy needs.

8.2.2. Surrogate Model for Additive Manufacturing

One of the broader goals of a ML-driven surrogate model is to be part of an interleaved FEM-ML simulation 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. Although the work in this thesis restricts itself to temperature profile prediction for an AM process, the same idea can be extended to related manufacturing processes such as incremental forming [236]. In general, this work can be extended to any phenomenon which utilizes partial differential equation based modeling.

Another possible future direction is to use a combination of recurrent neural network (RNN) and ensemble tree-based modeling. Stacked RNNs have been effective for learning the spatio-temporal nature of the additive manufacturing temperature profile [171]. However, RNNs take lot of time to train and would make it infeasible to be used directly in an iterative ML system. The RNN can be trained ex-situ on an existing dataset of FEM

simulations. Then, the penultimate layer of this ex-situ network can be used for generating features in-situ as part of the real-time system. Then, an ensemble tree algorithm can be applied on these set of features for an iterative real-time system.

Further, we suggest testing and benchmarking our approach across more complex geometries, different manufacturing parameters such as laser speed and intensity as well as FEM parameters such as across different mesh sizes.

8.2.3. Microstructure Optimization of Microstructures

The proposed approach for maximizing the yield stress under process constraints using data sampling algorithms can be extended for property optimizations for other non-linear design limitations and other materials. The sampling schemes are generalizable and agnostic of the problem domain and can be used in other scientific domains as well. There have been recent developments in reinforcement learning-based approaches for solving constrained optimization problems [237, 238]. In particular, multi-armed bandit based approaches have been used successfully for constrained optimization problems across domains [239]. A reinforcement learning-based approach would be automatically able to provide feedback to the sampling process without human guidance.

Another possible future direction is to consider Monte Carlo Tree Search-based methods instead of the proposed heuristic based tree-based models for reducing the constraint space. In addition, once we have selected one or more ODF solutions, it is possible to find similar solutions using Variational Auto-Encoders and Generative Adversarial Networks.

References

[1] Pınar Acar and Veeraraghavan Sundararaghavan. Utilization of a linear solver for multiscale design and optimization of microstructures in an airframe panel buckling problem. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 0156, 2016.

[2] Pinar Acar and Veera Sundararaghavan. A linear solution scheme for microstructure design with process constraints. *AIAA Journal*, 2016.

[3] Ilana Y Kanal, Steven G Owens, Jonathon S Bechtel, and Geoffrey R Hutchison. Efficient computational screening of organic polymer photovoltaics. *The journal of physical chemistry letters*, 4(10):1613–1623, 2013.

[4] 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.

[5] Tony Hey, Stewart Tansley, Kristin M Tolle, et al. *The fourth paradigm: data-intensive scientific discovery*, volume 1. Microsoft research Redmond, WA, 2009.

[6] Materials genome initiative, July 2016.

[7] 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.

[8] Logan Ward, Ankit Agrawal, Alok Choudhary, and Christopher Wolverton. A general-purpose machine learning framework for predicting properties of inorganic materials. *arXiv preprint arXiv:1606.09551*, 2016.

[9] Dezhen Xue, Prasanna V Balachandran, John Hogden, James Theiler, Deqing Xue, and Turab Lookman. Accelerated search for materials with targeted properties by adaptive design. *Nature communications*, 7, 2016.

[10] Venkatesh Botu and Rampi Ramprasad. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *International Journal of Quantum Chemistry*, 115(16):1074–1083, 2015.

[11] Ankit Agrawal, Parijat D Deshpande, Ahmet Cecen, Gautham P Basavarsu, Alok N Choudhary, and Surya R Kalidindi. Exploration of data science techniques to predict fatigue strength of steel from composition and processing parameters. *Integrating Materials and Manufacturing Innovation*, 3(1):1–19, 2014.

[12] Ruoqian Liu, Abhishek Kumar, Zhengzhang Chen, Ankit Agrawal, Veera Sundararaghavan, and Alok Choudhary. A predictive machine learning approach for microstructure optimization and materials design. *Scientific reports*, 5, 2015.

[13] HKDH Bhadeshia, RC Dimitriu, S Forsik, JH Pak, and JH Ryu. Performance of neural networks in materials science. *Materials Science and Technology*, 25(4):504–510, 2009.

[14] Edward O Pyzer-Knapp, Kewei Li, and Alan Aspuru-Guzik. Learning from the harvard clean energy project: The use of neural networks to accelerate materials discovery. *Advanced Functional Materials*, 25(41):6495–6502, 2015.

[15] Edward O Pyzer-Knapp, Gregor N Simm, and Alán Aspuru Guzik. A bayesian approach to calibrating high-throughput virtual screening results and application to organic photovoltaic materials. *Materials Horizons*, 3(3):226–233, 2016.

[16] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Muller, and O Anatole von Lilienfeld. Machine learning of molecular electronic properties in chemical compound space. *New Journal of Physics*, 15(9):095003, 2013.

[17] Bobby G Sumpter and Donald W Noid. On the design, analysis, and characterization of materials using computational neural networks. *Annual Review of Materials Science*, 26(1):223–277, 1996.

[18] Y Sun, WD Zeng, YQ Zhao, YL Qi, X Ma, and YF Han. Development of constitutive relationship model of ti600 alloy using artificial neural network. *Computational Materials Science*, 48(3):686–691, 2010.

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

[20] Eamonn Keogh and Abdullah Mueen. Curse of dimensionality. In *Encyclopedia of Machine Learning*, pages 257–258. Springer, 2011.

[21] 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.

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

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

[24] Victor E Kuz'min, Pavel G Polishchuk, Anatoly G Artemenko, and Sergey A Andronati. Interpretation of qsar models based on random forest methods. *Molecular informatics*, 30(6-7):593–603, 2011.

[25] XJ Yao, Annick Panaye, Jean-Pierre Doucet, RS Zhang, HF Chen, MC Liu, ZD Hu, and Bo Tao Fan. Comparative study of qsar/qspr correlations using support vector machines, radial basis function neural networks, and multiple linear regression. *Journal of chemical information and computer sciences*, 44(4):1257–1266, 2004.

[26] Adam G Gagorik, Brett Savoie, Nick Jackson, Ankit Agrawal, Alok Choudhary, Mark A Ratner, George C Schatz, and Kevin L Kohlstedt. Improved scaling of molecular network calculations: the emergence of molecular domains. *The journal of physical chemistry letters*, 8(2):415–421, 2017.

[27] Mojtaba Mozaffar, Arindam Paul, Reda Al-Bahrani, Sarah Wolff, Alok Choudhary, Ankit Agrawal, Kornel Ehmann, and Jian Cao. Data-driven prediction of the high-dimensional thermal history in directed energy deposition processes via recurrent neural networks. *Manufacturing Letters*, 18:35 – 39, 2018.

[28] Daylight Toolkit. Daylight chemical information systems. *Inc.: Aliso Viejo, CA*, 1997.

[29] Z Abdullah, Zainal Ahmad, and N Aziz. Multiple input-single output (miso) feedforward artificial neural network (fann) models for pilot plant binary distillation column. In *Bio-Inspired Computing: Theories and Applications (BIC-TA), 2011 Sixth International Conference on*, pages 157–160. IEEE, 2011.

[30] Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

[31] Glambard/molecules_dataset_collection: Collection of data sets of molecules for a validation of properties inference. `https://github.com/GLambard/Molecules_Dataset_`

Collection. (Accessed on 10/15/2018).

[32] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In *Advances in neural information processing systems*, pages 2224–2232, 2015.

[33] Garrett B Goh, Charles Siegel, Abhinav Vishnu, Nathan O Hodas, and Nathan Baker. Chemception: a deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models. *arXiv preprint arXiv:1706.06689*, 2017.

[34] Official repository for the chemixnet library. https://github.com/paularindam/CheMixNet. (Accessed on 11/01/2018).

[35] Mati Karelson, Victor S Lobanov, and Alan R Katritzky. Quantum-chemical descriptors in qsar/qspr studies. *Chemical reviews*, 96(3):1027–1044, 1996.

[36] Wendy A Warr. Representation of chemical structures. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(4):557–579, 2011.

[37] Mark A Johnson and Gerald M Maggiora. *Concepts and applications of molecular similarity*. Wiley, 1990.

[38] Malcolm J McGregor and Peter V Pallai. Clustering of large databases of compounds: Using the mdl "keys" as structural descriptors. *Journal of chemical information and computer sciences*, 37(3):443–448, 1997.

[39] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[40] Márton Vass, Albert J Kooistra, Tina Ritschel, Rob Leurs, Iwan JP de Esch, and Chris de Graaf. Molecular interaction fingerprint approaches for gpcr drug discovery. *Current Opinion in Pharmacology*, 30:59–68, 2016.

[41] Peter Willett. Similarity-based virtual screening using 2d fingerprints. *Drug discovery today*, 11(23):1046–1053, 2006.

[42] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of mdl keys for use in drug discovery. *Journal of chemical information and computer sciences*, 42(6):1273–1280, 2002.

[43] MACCS Structural Keys. Mdl information systems inc. *San Leandro, CA*, 2005.

[44] Garrett B Goh, Nathan O Hodas, Charles Siegel, and Abhinav Vishnu. Smiles2vec: An interpretable general-purpose deep neural network for predicting chemical properties. *arXiv preprint arXiv:1712.02034*, 2017.

[45] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[46] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[47] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[48] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[49] Alex Krizhevsky, I Sutskever, and G Hinton. Imagenet classification with deep convolutional neural. In *Neural Information Processing Systems*, pages 1–9, 2014.

[50] Junshui Ma, Robert P Sheridan, Andy Liaw, George E Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 55(2):263–274, 2015.

[51] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. Massively multitask networks for drug discovery. *arXiv preprint arXiv:1502.02072*, 2015.

[52] Guangyuan Kan, Cheng Yao, Qiaoling Li, Zhijia Li, Zhongbo Yu, Zhiyu Liu, Liuqian Ding, Xiaoyan He, and Ke Liang. Improving event-based rainfall-runoff simulation using an ensemble artificial neural network based hybrid data-driven model. *Stochastic environmental research and risk assessment*, 29(5):1345–1370, 2015.

[53] Garrett B Goh, Nathan Hodas, Charles Siegel, and Abhinav Vishnu. Smiles2vec: Predicting chemical properties from text representations. 2018.

[54] Cepdata.csv.zip. `https://www.dropbox.com/s/3kqzt9u1ryflls0/CEPData.csv.zip?dl=0`. (Accessed on 10/15/2018).

[55] Wichien Sang-aroon, Seksan Laopha, Phrompak Chaiamornnugool, Sarawut Tontapha, Samarn Saekow, and Vittaya Amornkitbamrung. Dft and tddft study on the electronic structure and photoelectrochemical properties of dyes derived from cochineal and lac insects as photosensitizer for dye-sensitized solar cells. *Journal of molecular modeling*, 19(3):1407–1415, 2013.

[56] Zhong Hu, Vedbar S Khadka, Wei Wang, David W Galipeau, and Xingzhong Yan. Theoretical study of two-photon absorption properties and up-conversion efficiency of new symmetric organic π-conjugated molecules for photovoltaic devices. *Journal of molecular modeling*, 18(8):3657–3667, 2012.

[57] Mazmira Mohamad, Rashid Ahmed, Amirudin Shaari, and Souraya Goumri-Said. First principles investigations of vinazene molecule and molecular crystal: a prospective candidate for organic photovoltaic applications. *Journal of molecular modeling*, 21(2):27, 2015.

[58] Natalia Inostroza, Fernando Mendizabal, Ramiro Arratia-Pérez, Carlos Orellana, and Cristian Linares-Flores. Improvement of photovoltaic performance by substituent effect of

donor and acceptor structure of tpa-based dye-sensitized solar cells. *Journal of molecular modeling*, 22(1):25, 2016.

[59] Claudia N Hoth, Roland Steim, Pavel Schilinsky, Stelios A Choulis, Sandro F Tedde, Oliver Hayden, and Christoph J Brabec. Topographical and morphological aspects of spray coated organic photovoltaics. *Organic Electronics*, 10(4):587–593, 2009.

[60] Markus C Scharber, David Mühlbacher, Markus Koppe, Patrick Denk, Christoph Waldauf, Alan J Heeger, and Christoph J Brabec. Design rules for donors in bulk-heterojunction solar cells—towards 10% energy-conversion efficiency. *Advanced materials*, 18(6):789–794, 2006.

[61] François Chollet et al. Keras, 2015.

[62] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[63] Greg Landrum. Rdkit: Open-source cheminformatics. 3(04):2012, 2006.

[64] 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.

[65] deepchem/deepchem: Democratizing deep-learning for drug discovery, quantum chemistry, materials science and biology. `https://github.com/deepchem/deepchem`.

[66] maxpumperla/hyperas: Keras + hyperopt: A very simple wrapper for convenient hyperparameter optimization. `https://github.com/maxpumperla/hyperas`. (Accessed on 10/16/2018).

[67] hyperopt/hyperopt: Distributed asynchronous hyperparameter optimization in python. `https://github.com/hyperopt/hyperopt`. (Accessed on 10/16/2018).

[68] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[69] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

[70] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.

[71] Rosaria Ciriminna, Francesco Meneguzzo, Mario Pecoraino, and Mario Pagliaro. Rethinking solar energy education on the dawn of the solar economy. *Renewable and Sustainable Energy Reviews*, 63:13–18, 2016.

[72] MC Scharber and NS Sarciftci. Bulk heterojunction organic solar cells: Working principles and power conversion efficiencies. *Nanostructured Materials for Type III Photovoltaics*, 45:33, 2017.

[73] Jiangeng Xue, Soichi Uchida, Barry P Rand, and Stephen R Forrest. Asymmetric tandem organic photovoltaic cells with hybrid planar-mixed molecular heterojunctions. *Applied Physics Letters*, 85(23):5757–5759, 2004.

[74] Jianhui Hou and Xia Guo. Active layer materials for organic solar cells. In *Organic Solar Cells*, pages 17–42. Springer, 2013.

[75] P Granero, VS Balderrama, J Ferré-Borrull, J Pallarès, and LF Marsal. Two-dimensional finite-element modeling of periodical interdigitated full organic solar cells. *Journal of Applied Physics*, 113(4):043107, 2013.

[76] Yongjeong Lee, Kyungnam Kang, Sanghwa Lee, Hyeong Pil Kim, Jin Jang, and Jungho Kim. Integrated optoelectronic model for organic solar cells based on the finite element

method including the effect of oblique sunlight incidence and a non-ohmic electrode contact. *Japanese Journal of Applied Physics*, 55(10):102301, 2016.

[77] Warren J. Hehre. *Ab initio molecular orbital theory*. Wiley-Interscience, 1986.

[78] Jeremy Taylor, Hong Guo, and Jian Wang. Ab initio modeling of quantum transport properties of molecular electronic devices. *Physical Review B*, 63(24):245407, 2001.

[79] Jean-Luc Brédas, Joseph E Norton, Jérôme Cornil, and Veaceslav Coropceanu. Molecular understanding of organic solar cells: the challenges. *Accounts of chemical research*, 42(11):1691–1699, 2009.

[80] Abraham Yosipof, Omer Kaspi, Koushik Majhi, and Hanoch Senderowitz. Visualization based data mining for comparison between two solar cell libraries. *Molecular informatics*, 35(11-12):622–628, 2016.

[81] Peter B Jørgensen, Mikkel N Schmidt, and Ole Winther. Deep generative models for molecular science. *Molecular informatics*, 37(1-2):1700133, 2018.

[82] Omer Kaspi, Abraham Yosipof, and Hanoch Senderowitz. Pv analyzer: A decision support system for photovoltaic solar cells libraries. *Molecular informatics*, 37(9-10):1800067, 2018.

[83] Jean Roncali, Philippe Leriche, and Philippe Blanchard. Molecular materials for organic photovoltaics: small is beautiful. *Advanced Materials*, 26(23):3821–3838, 2014.

[84] Sam-Shajing Sun and Niyazi Serdar Sariciftci. *Organic photovoltaics: mechanisms, materials, and devices*. CRC press, 2017.

[85] Sarah Holliday, Yilin Li, and Christine Luscombe. Recent advances in high performance donor-acceptor polymers for organic photovoltaics. *Progress in Polymer Science*, 2017.

[86] Johannes Hachmann, Roberto Olivares-Amaya, Sule Atahan-Evrenk, Carlos Amador-Bedolla, Roel S Sánchez-Carrera, Aryeh Gold-Parker, Leslie Vogt, Anna M Brockway, and Alán Aspuru-Guzik. The harvard clean energy project: large-scale computational screening and design of organic photovoltaics on the world community grid. *The Journal*

of Physical Chemistry Letters, 2(17):2241–2251, 2011.

[87] Harikrishna Sahu, Weining Rao, Alessandro Troisi, and Haibo Ma. Toward predicting efficiency of organic solar cells via machine learning and improved descriptors. *Advanced Energy Materials*, 8(24):1801032, 2018.

[88] 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.

[89] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547, 2018.

[90] Arindam Paul, Pinar Acar, Ruoqian Liu, Wei-Keng Liao, Alok Choudhary, Veera Sundararaghavan, and Ankit Agrawal. Data sampling schemes for microstructure design with vibrational tuning constraints. *AIAA Journal*, 56(3):1239–1250, 2018.

[91] Peter Bjørn Jørgensen, Murat Mesta, Suranjan Shil, Juan Maria García Lastra, Karsten Wedel Jacobsen, Kristian Sommer Thygesen, and Mikkel N Schmidt. Machine learning-based screening of complex molecules for polymer solar cells. *The Journal of chemical physics*, 148(24):241735, 2018.

[92] 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.

[93] Bing Cao, Lawrence A Adutwum, Anton O Oliynyk, Erik J Luber, Brian C Olsen, Arthur Mar, and Jillian M Buriak. How to optimize materials and devices via design of experiments and machine learning: Demonstration using organic photovoltaics. *ACS nano*, 12(8):7434–7444, 2018.

[94] Dipendra Jha, Logan Ward, Zijiang Yang, Christopher Wolverton, Ian Foster, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Irnet: A general purpose deep residual regression framework for materials discovery. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019.

[95] Dipendra Jha, Aaron Gilad Kusne, Nam Nguyen, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Peak area detection network for directly learning phase regions from raw x-ray diffraction patterns. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019.

[96] Zijiang Yang, Yuksel C Yabansu, Dipendra Jha, Wei-keng Liao, Alok N Choudhary, Surya R Kalidindi, and Ankit Agrawal. Establishing structure-property localization linkages for elastic deformation of three-dimensional high contrast composites using deep learning approaches. *Acta Materialia*, 166:335–345, 2019.

[97] Ankit Agrawal and Alok Choudhary. Deep materials informatics: Applications of deep learning in materials science. *MRS Communications*, pages 1–14, 2019.

[98] Juan-Pablo Correa-Baena, Kedar Hippalgaonkar, Jeroen van Duren, Shaffiq Jaffer, Vijay R Chandrasekhar, Vladan Stevanovic, Cyrus Wadia, Supratik Guha, and Tonio Buonassisi. Accelerating materials development via automation, machine learning, and high-performance computing. *Joule*, 2(8):1410–1420, 2018.

[99] Adam C Mater and Michelle L Coote. Deep learning in chemistry. *Journal of chemical information and modeling*, 2019.

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

[101] Omer Kaspi, Abraham Yosipof, and Hanoch Senderowitz. Random sample consensus (ransac) algorithm for material-informatics: application to photovoltaic solar cells. *Journal of cheminformatics*, 9(1):34, 2017.

[102] Johannes Hachmann, Roberto Olivares-Amaya, Adrian Jinich, Anthony L Appleton, Martin A Blood-Forsythe, Laszlo R Seress, Carolina Roman-Salgado, Kai Trepte, Sule Atahan-Evrenk, Süleyman Er, et al. Lead candidates for high-performance organic photovoltaics from high-throughput quantum chemistry–the harvard clean energy project. *Energy & Environmental Science*, 7(2):698–704, 2014.

[103] 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.

[104] 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.

[105] Markus Clark Scharber and Niyazi Serdar Sariciftci. Efficiency of bulk-heterojunction organic solar cells. *Progress in polymer science*, 38(12):1929–1940, 2013.

[106] Satya Avasarala. *Selenium WebDriver practical guide*. Packt Publishing Ltd, 2014.

[107] Leonard Richardson. Beautiful soup documentation. *April*, 2007.

[108] The harvard organic photovoltaics 2015 (hopv) dataset: An experiment-theory calibration resource. `https://figshare.com/articles/HOPV15_Dataset/1610063`, 2016. (Accessed on 09/22/2016).

[109] O Anatole von Lilienfeld, Raghunathan Ramakrishnan, Matthias Rupp, and Aaron Knoll. Fourier series of atomic radial distribution functions: A molecular fingerprint for machine learning models of quantum chemical properties. *International Journal of Quantum Chemistry*, 115(16):1084–1093, 2015.

[110] Florian Weigend and Reinhart Ahlrichs. Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for h to rn: Design and assessment of accuracy. *Physical Chemistry Chemical Physics*, 7(18):3297–3305, 2005.

[111] Iiris Kahn, Andre Lomaka, and Mati Karelson. Topological fingerprints as an aid in finding structural patterns for lrrk2 inhibition. *Molecular informatics*, 33(4):269–275, 2014.

[112] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63, 2015.

[113] Sereina Riniker and Gregory A Landrum. Open-source platform to benchmark fingerprints for ligand-based virtual screening. *Journal of cheminformatics*, 5(1):1, 2013.

[114] Andreas Bender and Robert C Glen. Molecular similarity: a key technique in molecular informatics. *Organic & biomolecular chemistry*, 2(22):3204–3218, 2004.

[115] Michael Reutlinger, Christian P Koch, Daniel Reker, Nickolay Todoroff, Petra Schneider, Tiago Rodrigues, and Gisbert Schneider. Chemically advanced template search (cats) for scaffold-hopping and prospective target prediction for orphan molecules. *Molecular informatics*, 32(2):133–138, 2013.

[116] Yasuo Tabei and Koji Tsuda. Sketchsort: Fast all pairs similarity search for large databases of molecular fingerprints. *Molecular informatics*, 30(9):801–807, 2011.

[117] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

[118] G Landrum. Rdkit: open-source cheminformatics software, 2016.

[119] Navpreet Kaur, Mandeep Singh, Dinesh Pathak, Tomas Wagner, and JM Nunzi. Organic materials for photovoltaic applications: Review and mechanism. *Synthetic Metals*, 190:20–26, 2014.

[120] Yuze Lin and Xiaowei Zhan. Non-fullerene acceptors for organic photovoltaics: an emerging horizon. *Materials Horizons*, 1(5):470–488, 2014.

[121] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[122] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[123] 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.

[124] Thomas F Stocker, Dahe Qin, Gian-Kasper Plattner, M Tignor, Simon K Allen, Judith Boschung, Alexander Nauels, Yu Xia, Vincent Bex, and Pauline M Midgley. Climate change 2013: The physical science basis, 2014.

[125] Andrew Hoffman. Computational chemistry in rational material design for organic photovoltaics. 2015.

[126] Godfrey Boyle et al. *Renewable energy: power for a sustainable future*. Taylor & Francis, 1997.

[127] John A Turner. A realizable renewable energy future. *Science*, 285(5428):687–689, 1999.

[128] Larry Baxter. Biomass-coal co-combustion: opportunity for affordable renewable energy. *Fuel*, 84(10):1295–1302, 2005.

[129] NASA. Nasa - clean energy. `https://www.nasa.gov/centers/ames/greenspace/clean-energy.html`, 2016.

[130] Junsheng Yu, Yifan Zheng, and Jiang Huang. Towards high performance organic photovoltaic cells: A review of recent development in organic photovoltaics. *Polymers*, 6(9):2473–2509, 2014.

[131] Christoph Brabec, Ullrich Scherf, and Vladimir Dyakonov. *Organic photovoltaics: materials, device physics, and manufacturing technologies*. John Wiley & Sons, 2011.

[132] Omar A Abdulrazzaq, Viney Saini, Shawn Bourdo, Enkeleda Dervishi, and Alexandru S Biris. Organic solar cells: a review of materials, limitations, and possibilities for improvement. *Particulate science and technology*, 31(5):427–442, 2013.

[133] Stephen R Forrest. The limits to organic photovoltaic cell efficiency. *MRS bulletin*, 30(1):28–32, 2005.

[134] Sinno Jialin Pan, Qiang Yang, et al. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

[135] Shin Hoo-Chang, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285, 2016.

[136] Stephen R Heller, Alan McNaught, Igor Pletnev, Stephen Stein, and Dmitrii Tchekhovskoi. Inchi, the iupac international chemical identifier. *Journal of cheminformatics*, 7(1):23, 2015.

[137] Giampaolo Barone, Dario Duca, Arturo Silvestri, Luigi Gomez-Paloma, Raffaele Riccio, and Giuseppe Bifulco. Determination of the relative stereochemistry of flexible organic compounds by ab initio methods: conformational analysis and boltzmann-averaged giao 13c nmr chemical shifts. *Chemistry–A European Journal*, 8(14):3240–3245, 2002.

[138] 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.

[139] 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.

[140] 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.

[141] 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.

[142] 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.

[143] 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.

[144] 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.

[145] 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.

[146] C Li, ZY Liu, XY Fang, and YB Guo. Residual stress in metal additive manufacturing. *Procedia Cirp*, 71:348–353, 2018.

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

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

[149] 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.

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

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

[152] 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.

[153] 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.

[154] 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.

[155] 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.

[156] 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.

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

[158] Michael E Stender, Lauren L Beghini, Joshua D Sugar, Michael G Veilleux, Samuel R Subia, Thale R Smith, Christopher W San Marchi, Arthur A Brown, and Daryl J Dagel. A thermal-mechanical finite element workflow for directed energy deposition additive manufacturing process modeling. *Additive Manufacturing*, 21:556–566, 2018.

[159] 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.

[160] 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.

[161] 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.

[162] 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.

[163] Arindam Paul, Alona Furmanchuk, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. Property prediction of organic donor molecules for photovoltaic applications using extremely randomized trees. *Molecular Informatics*, 2019.

[164] 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.

[165] 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.

[166] 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.

[167] 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.

[168] Arindam Paul, Pinar Acar, Ruoqian Liu, Wei-keng Liao, Alok Choudhary, Veera Sundararaghavan, and Ankit Agrawal. Data sampling schemes for microstructure design with vibrational tuning constraints. *AIAA Journal*, 56(3):1239–1250, 2018.

[169] 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.

[170] 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*,

160:334351, 2019.

[171] Mojtaba Mozaffar, Arindam Paul, Reda Al-Bahrani, Sarah Wolff, Alok Choudhary, Ankit Agrawal, Kornel Ehmann, and Jian Cao. Data-driven prediction of the high-dimensional thermal history in directed energy deposition processes via recurrent neural networks. *Manufacturing letters*, 18:35–39, 2018.

[172] 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.

[173] 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.

[174] 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.

[175] 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.

[176] 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.

[177] 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.

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

[179] 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.

[180] 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.

[181] 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.

[182] 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.

[183] 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.

[184] Ruoqian Liu, Yuksel C Yabansu, Zijiang Yang, Alok N Choudhary, Surya R Kalidindi, and Ankit Agrawal. Context aware machine learning approaches for modeling elastic localization in three-dimensional composite microstructures. *Integrating Materials and Manufacturing Innovation*, pages 1–12, 2017.

[185] PD Deshpande, BP Gautham, A Cecen, S Kalidindi, Ankit Agrawal, and A Choudhary. Application of statistical and machine learning techniques for correlating properties to composition and manufacturing processes of steels. In *Proceedings of the 2nd World Congress on Integrated Computational Materials Engineering (ICME)*, pages 155–160. Springer, 2013.

[186] Ruoqian Liu, Ankit Agrawal, Wei-keng Liao, Alok Choudhary, and Marc De Graef. Materials discovery: understanding polycrystals from large-scale electron patterns. In *Big Data*

(Big Data), 2016 IEEE International Conference on, pages 2261–2269. IEEE, 2016.

[187] Ruoqian Liu, Yuksel C Yabansu, Ankit Agrawal, Surya R Kalidindi, and Alok N Choudhary. Machine learning approaches for elastic localization linkages in high-contrast composite materials. *Integrating Materials and Manufacturing Innovation*, 4(1):13, 2015.

[188] Abdul-Ghani Olabi and Artur Grunwald. Design and application of magnetostrictive materials. *Materials & Design*, 29(2):469–483, 2008.

[189] R Grossinger, R Sato Turtelli, and N Mehmood. Materials with high magnetostriction. In *IOP Conference Series: Materials Science and Engineering*, volume 60, page 012002. IOP Publishing, 2014.

[190] Frank Claeyssen, N Lhermet, R Le Letty, and P Bouchilloux. Actuators, transducers and motors based on giant magnetostrictive materials. *Journal of alloys and compounds*, 258(1-2):61–73, 1997.

[191] Heng Zhang, Tianli Zhang, and Chengbao Jiang. Design of a uniform bias magnetic field for giant magnetostrictive actuators applying triple-ring magnets. *Smart Materials and Structures*, 22(11):115009, 2013.

[192] Yoshio Yamamoto, Hiroshi Eda, and Jun Shimizu. Application of giant magnetostrictive materials to positioning actuators. In *Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on*, pages 215–220. IEEE, 1999.

[193] Abhishek Kumar and Veera Sundararaghavan. Simulation of magnetostrictive properties of galfenol under thermomechanical deformation. *Finite Elements in Analysis and Design*, 127:1–5, 2017.

[194] RA Kellogg, AM Russell, TA Lograsso, AB Flatau, AE Clark, and M Wun-Fogle. Tensile properties of magnetostrictive iron–gallium alloys. *Acta Materialia*, 52(17):5043–5050, 2004.

[195] JB Restorff, M Wun-Fogle, and AE Clark. Measurement of d 15 in fe 100- x ga x (x= 12.5, 15, 18.4, 22), fe 50 co 50, and fe 81 al 19 highly textured polycrystalline rods. *Journal of Applied Physics*, 103(7):07B305, 2008.

[196] A Mahadevan, PG Evans, and MJ Dapino. Dependence of magnetic susceptibility on stress in textured polycrystalline fe 81.6 ga 18.4 and fe 79.1 ga 20.9 galfenol alloys. *Applied Physics Letters*, 96(1):012502, 2010.

[197] Leon M Cheng, Allison E Nolting, Benoit Voyzelle, and Claude Galvani. Deformation behavior of polycrystalline galfenol at elevated temperatures. *Behavior and Mechanics of Multifunctional and Composite Materials, Edited by Dapino, Marcelo J.. Proceedings of the SPIE*, 6526:65262N, 2007.

[198] Suok-Min Na and Alison B Flatau. Deformation behavior and magnetostriction of polycrystalline fe–ga–x (x= b, c, mn, mo, nb, nb c) alloys. *Journal of Applied Physics*, 103(7):07D304, 2008.

[199] N Srisukhumbowornchai and S Guruswamy. Crystallographic textures in rolled and annealed fe-ga and fe-al alloys. *Metallurgical and Materials Transactions A*, 35(9):2963–2970, 2004.

[200] Brent L Adams, A Henrie, B Henrie, M Lyon, SR Kalidindi, and H Garmestani. Microstructure-sensitive design of a compliant beam. *Journal of the Mechanics and Physics of Solids*, 49(8):1639–1663, 2001.

[201] Surya R Kalidindi, Joshua R Houskamp, Mark Lyons, and Brent L Adams. Microstructure sensitive design of an orthotropic plate subjected to tensile load. *International Journal of Plasticity*, 20(8):1561–1575, 2004.

[202] Tony Fast, Marko Knezevic, and Surya R Kalidindi. Application of microstructure sensitive design to structural components produced from hexagonal polycrystalline metals. *Computational Materials Science*, 43(2):374–383, 2008.

[203] Pinar Acar and Veera Sundararaghavan. Utilization of a linear solver for multiscale design and optimization of microstructures. *AIAA Journal*, pages 1751–1759, 2016.

[204] Suok-Min Na and Alison B Flatau. Secondary recrystallization, crystallographic texture and magnetostriction in rolled fe–ga based alloys. *Journal of applied physics*, 101(9):09N518, 2007.

[205] H-J Bunge. *Texture analysis in materials science: mathematical methods*. Elsevier, 2013.

[206] A Heinz and P Neumann. Representation of orientation and disorientation data for cubic, hexagonal, tetragonal and orthorhombic crystals. *Acta Crystallographica Section A: Foundations of Crystallography*, 47(6):780–789, 1991.

[207] U Fred Kocks, Carlos Norberto Tomé, and H-R Wenk. *Texture and anisotropy: preferred orientations in polycrystals and their effect on materials properties*. Cambridge university press, 2000.

[208] Valerie Randle and Olaf Engler. *Introduction to texture analysis: macrotexture, microtexture and orientation mapping*. CRC press, 2000.

[209] A Kumar and PR Dawson. Computational modeling of fcc deformation textures over rodrigues' space. *Acta Materialia*, 48(10):2719–2736, 2000.

[210] Geoffrey Ingram Taylor. Analysis of plastic strain in a cubic crystal. *Stephen Timoshenko 60th Anniversary Volume*, pages 218–224, 1938.

[211] DH Chung and WR Buessem. The elastic anisotropy of crystals. *Journal of Applied Physics*, 38(5):2010–2012, 1967.

[212] Hui Wang, YN Zhang, RQ Wu, LZ Sun, DS Xu, and ZD Zhang. Understanding strong magnetostriction in fe100- xgax alloys. *Scientific reports*, 3, 2013.

[213] R Shi, N Zhou, SR Niezgoda, and Y Wang. Microstructure and transformation texture evolution during α precipitation in polycrystalline α/β titanium alloys–a simulation study. *Acta Materialia*, 94:224–243, 2015.

[214] Philip Weetman and George Akhras. Modeling a galfenol based stress sensor capable of sensing up to three axial stresses. *Journal of Applied Physics*, 114(18):183911, 2013.

[215] AJ Boesenberg, JB Restorff, M Wun-Fogle, H Sailsbury, and E Summers. Texture development in galfenol wire. *Journal of Applied Physics*, 113(17):17A909, 2013.

[216] JP Domann, CM Loeffler, BE Martin, and GP Carman. High strain-rate magnetoelasticity in galfenol. *Journal of Applied Physics*, 118(12):123904, 2015.

[217] H Cao, Peter M Gehring, CP Devreugd, JA Rodriguez-Rivera, J Li, and D Viehland. Role of nanoscale precipitates on the enhanced magnetostriction of heat-treated galfenol (fe 1-x ga x) alloys. *Physical review letters*, 102(12):127201, 2009.

[218] Rick Allen Kellogg. *Development and modeling of iron-gallium alloys*. PhD thesis, Iowa State University, 2003.

[219] PR Downey and AB Flatau. Magnetoelastic bending of galfenol for sensor applications. *Journal of Applied Physics*, 97(10):10R505, 2005.

[220] Supratik Datta and Alison B Flatau. Magnetostrictive vibration sensor based on iron-gallium alloy. In *MRS Proceedings*, volume 888, pages 0888–V04. Cambridge Univ Press, 2005.

[221] Ruoqian Liu, Ankit Agrawal, Wei-keng Liao, Alok Choudhary, and Zhengzhang Chen. Pruned search: A machine learning based meta-heuristic approach for constrained continuous optimization. In *Contemporary Computing (IC3), 2015 Eighth International Conference on*, pages 13–18. IEEE, 2015.

[222] Virginia Torczon. On the convergence of pattern search algorithms. *SIAM Journal on optimization*, 7(1):1–25, 1997.

[223] Hongyi Xu, Ruoqian Liu, Alok Choudhary, and Wei Chen. A machine learning-based design representation method for designing heterogeneous microstructures. *Journal of Mechanical Design*, 137(5):051403, 2015.

[224] William D Brewer, R Keith Bird, and Terryl A Wallace. Titanium alloys and processing for high speed aircraft. *Materials Science and Engineering: A*, 243(1-2):299–304, 1998.

[225] Valentin N Moiseyev. *Titanium alloys: Russian aircraft and aerospace applications*. CRC press, 2005.

[226] AG Bratukhin, BA Kolachev, VV Sadkov, et al. Technology of production of titanium aircraft structures. *Mashinostroenie, Moscow*, 1995.

[227] RR Boyer. Titanium for aerospace: rationale and applications. *Advanced Performance Materials*, 2(4):349–368, 1995.

[228] AR Machado and J Wallbank. Machining of titanium and its alloys–a review. 1989.

[229] EO Ezugwu and ZM Wang. Titanium alloys and their machinability a review. *Journal of Materials Processing Technology*, 68:262–274, 1997.

[230] RV Grandhi, SC Modukuru, and JC Malas. Integrated strength and manufacturing process design using a shape optimization approach. *Journal of Mechanical Design*, 115(1):125–131, 1993.

[231] Hongyi Xu, Yang Li, Catherine Brinson, and Wei Chen. A descriptor-based design methodology for developing heterogeneous microstructural materials system. *Journal of Mechanical Design*, 136(5):051007, 2014.

[232] Arindam Paul, Pinar Acar, Ruoqian Liu, Wei-keng Liao, Alok Choudhary, Veera Sundararaghavan, and Ankit Agrawal. Data sampling schemes for microstructure design with vibrational tuning constraints. *AIAA Journal*, 56(3):1239–1250, 2018.

[233] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[234] Thais Mayumi Oshiro, Pedro Santoro Perez, and José Augusto Baranauskas. How many trees in a random forest? In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 154–168. Springer, 2012.

[235] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

[236] Suresh Kurra, Nasih Hifzur Rahman, Srinivasa Prakash Regalla, and Amit Kumar Gupta. Modeling and optimization of surface roughness in single point incremental forming process. *Journal of Materials Research and Technology*, 4(3):304–313, 2015.

[237] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 22–31. JMLR. org, 2017.

[238] Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

[239] Avinash Balakrishnan, Djallel Bouneffouf, Nicholas Mattei, and Francesca Rossi. Using contextual bandits with behavioral constraints for constrained online movie recommendation. In *IJCAI*, pages 5802–5804, 2018.