DopeDefects

Predicting impurity energy levels of semiconductors using machine learning

Ryan Beck, Lauren Koulias, Linnette Teo
Project Mentors: Argonne National Lab – Maria K. Chan, Arun Kumar Mannodi Kanakkithodi

Overview

Overview: DopeDefects is an open source python package that aims to predict the enthalpy of formations, as well as the charge transition levels, of various defects embedded in Cd/chalcogenide crystals.

Available on GitHub:

https://github.com/dopedefects/dopedefects.git

Motivation

- Chemical space for potential solar cell materials is large
- Use Density Functional Theory (DFT) computations, an ab inito method for calculating chemical properties
- DFT requires significant computational resources both in time and energy costs
- Number of calculations required to explore entire space is unfeasible
- Possible solution: predictive models trained on small subset of calculated properties

About the Data

Properties to predict

- Supercell enthalpies of formation (3)
- Supercell energies of charged states (6)

Descriptors of defect system (109 in total)

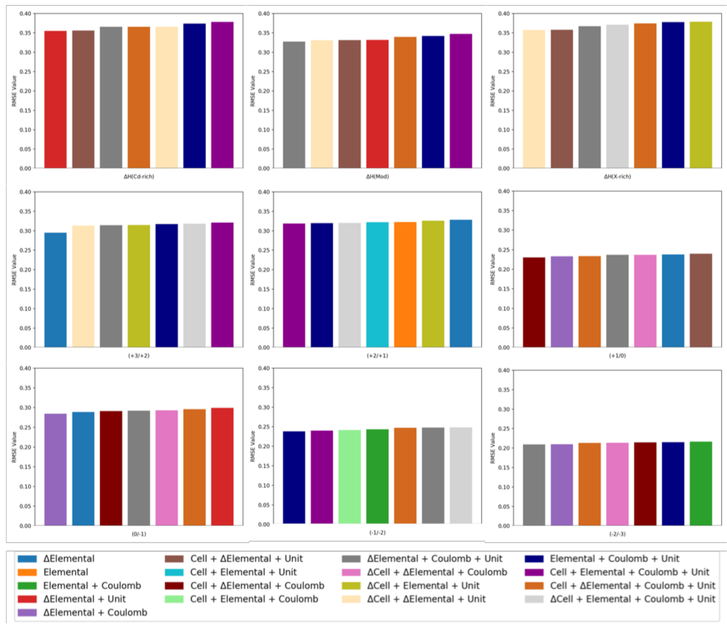
- ***Elemental:*** elemental properties of the dopants such as group, period, ionic, atomic, & covalent radii, boiling point, atomic weight, etc.
- ***ΔElemental:*** change in elemental properties between the dopant atom and the atom that it replaced
- ***Unit:*** ΔH values from the unit cell calculation, conduction and valence band edges
- ***Cell:*** bond angles and bond lengths for all atoms in the unit cell
- ***ΔCell:*** change in the bond angles and bond lengths for all atoms between the doped and undoped cell
- ***Coulomb:*** coulomb matrix for the unit cell

Data Cleaning Functionalities

- Scan through the provided directory for VASP (Vienna Ab inito Simulation Package) geometry files and convert the coordinates to cartesian space
- Determine the position and type of vacancy
- Calculate the bond lengths and angles for the atoms surrounding the defect, as well as determining the change in comparison to a pure system
- Collect all the properties into a pandas dataframe, as well as save and resume the data so that data parsing does not need to be redone

Feature Selection

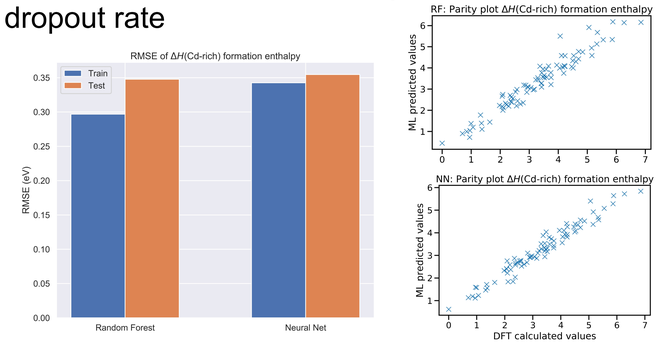
RMSE for Random Forrest Regression



- For each property being predicted, a different set of descriptors was the most accurate, the top 7 for each category are shown above
- Overall it seems that Elemental properties are always necessary, combined with other descriptors for the most accurate results

Neural networks

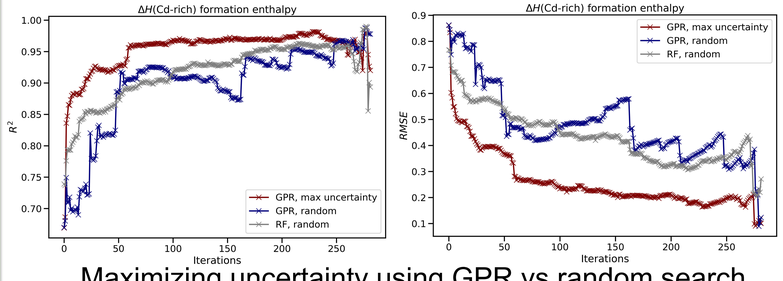
- Used elemental and unit descriptors; 425 data points
- Split data 6:2:2 (training:validation:testing)
- Hyperparameter tuning: batch size, epoch number, number of nodes in hidden layers, learning rate, dropout rate



Neural net does not perform significantly better than random forest – need further optimization, more data

Iterative Method using Gaussian Process Regression

- Start with small subset of data to fit GPR model (10% of 315 CdTe structures)
- Use model to predict mean and uncertainty (standard deviation) on remaining test points
- Choose a test point that maximizes uncertainty
- Add test point (with calculated value) to model and retrain
- Iterate - keep adding points till satisfied



Maximizing uncertainty using GPR vs random search helps reduce initial number of known points needed

Future Work

- Multiple output predictions
- Improvement of prediction for impurity transition levels
- More detailed analysis into different CdX structures