DFT Calculations, ML Predictions of Si-Metal Alloy Potentials

Course Project Report for **ME438** (Autumn 2024-25)

Yash Salunkhe
Department of Mechanical Engg,
210020156
yash.salunkhe@iitb.ac.in

***Abstract*—Silicon-metal alloys play a crucial role in advanced material applications, which requires precise modeling of their interatomic potentials to improve understanding of their properties. In this work, Density Functional Theory (DFT) was used to compute energy profiles and potentials for Si-metal systems, including Li, Na, K, Mg, Ca, and pure Si. To accelerate and improve potential prediction, three machine learning (ML) techniques were employed: Support Vector Regression (SVR), Gaussian Mixture Models (GMM), and Fully Connected Neural Networks (FCNN). These models were trained on DFT-generated data to predict alloy potentials efficiently. The integration of DFT and ML provides a robust framework for accurate and computationally efficient potential modeling, enabling rapid material property exploration and design.**

I. Introduction

Silicon-metal alloys are of significant interest due to their unique mechanical, thermal, and electronic properties, which make them essential for a wide range of applications, including semiconductors, energy storage, and structural materials. Understanding the interatomic interactions within these alloys is critical for predicting their thermodynamic stability, mechanical behavior, and potential performance in real-world applications.

Traditionally, Density Functional Theory (DFT) has been the method of choice for computing interatomic potentials and energy profiles with high accuracy. However, DFT calculations are computationally intensive, particularly for complex alloy systems or large-scale simulations. To address this challenge, integrating Machine Learning (ML) techniques offers a promising solution by providing a means to approximate these potentials with significantly reduced computational cost while maintaining accuracy.

In this project, DFT was employed to calculate energy profiles for various Si-metal systems, including alkali metals (Li, Na, K), alkaline earth metals (Mg, Ca), and pure Silicon. Using the resulting DFT data, three distinct ML approaches were developed and tested:

- Support Vector Regression (SVR): A robust and efficient technique for modeling complex relationships between atomic configurations and potentials.
- Gaussian Mixture Models (GMMs): A probabilistic approach capable of capturing the distribution of energy surfaces across alloy systems.
- Fully Connected Neural Networks (FCNNs): A deep learning method designed to extract high-order features and provide highly accurate potential predictions.

The objective of this work is to demonstrate how ML models can be trained on DFT-generated datasets to predict interatomic potentials efficiently and accurately, enabling faster exploration and optimization of material properties. This integration of DFT and ML serves as a powerful framework for advancing materials science and engineering by reducing computational costs while maintaining high fidelity in simulations.

II. Dataset Generation and Processing

A. Data Collection

The dataset is focused on material property data for Si-based and Na-based alloys, obtained from the Materials Project database using the `MPRester` class from the `pymatgen` library. The query was structured to retrieve materials containing specific elements (e.g., Si, Na), and properties of interest include:

- `material_id`: Unique identifier for each material.
- `pretty_formula`: Chemical formula of the material.
- `energy_per_atom`: Energy per atom of the material.
- `structure`: Structural information of the material.
- `e_above_hull`: Energy above the convex hull.
- `band_gap`: Band gap of the material.
- `formation_energy_per_atom`: Formation energy per atom.
- `crystal_system`: Crystal system of the material.
- `energy`: Total energy of the material.

B. Data Processing

The collected data was processed and structured into a `pandas` DataFrame for further analysis. Additional steps included:

- Adding categorical columns such as `Category_label` (e.g., 1 for Li-based, 2 for Na-based) and `Class_name` (e.g., ”Li_based”, ”Na_based”).

- Using `matminer` to perform feature engineering on the material structures.
- Performing Density Functional Theory (DFT) calculations for a subset of materials to generate potential energy surfaces.

C. Feature Engineering and DFT Calculations

The feature engineering and computational analysis involved the following steps:

1) **Density Features (Sine)**: Calculated using the `DensityFeatures` class to extract density-related characteristics.
2) **XRD Powder Pattern**: Generated using the `XRDPowderPattern` class to simulate X-ray diffraction patterns.
3) **Orbital Field Matrix**: Computed using the `OrbitalFieldMatrix` class to derive orbital-based features from material structures.
4) **DFT Calculations**: Using the PySCF package, density functional theory (DFT) computations were performed to calculate accurate potential energy surfaces for alloy systems.

D. Generated Datasets

Four distinct datasets were produced and saved as CSV files:

- **Material Properties**: Includes properties like energy per atom, formation energy per atom, and band gap.
- **Categorical Data**: Contains labels and class names categorizing materials by type.
- **Featurized Data**: Includes density features, XRD powder patterns, and orbital field matrices.
- **DFT Results**: Contains detailed energy values and potential energy surface data for Si-based and Na-based alloys.

E. Summary

The dataset is a comprehensive collection of material properties, engineered features, and computed energy data. Advanced feature extraction techniques and DFT computations enable a rich representation of material structures, suitable for further analysis and machine learning tasks.

III. Model Details

A. Gaussian Mixture Models (GMMs)

Gaussian Mixture Models (GMMs) are probabilistic models that represent data as a mixture of multiple Gaussian distributions. Each Gaussian component is characterized by a mean, variance, and a mixing coefficient. GMMs were used in this project to identify patterns in the featurized dataset derived from Density Functional Theory (DFT) calculations. They modeled the probabilistic distributions of potential energy surfaces for Si-metal systems, capturing multi-modal behavior across different alloy configurations.

B. Support Vector Regression (SVR)

Support Vector Regression (SVR) is a supervised learning algorithm that predicts continuous outputs by fitting a regression line within a margin of tolerance (ϵ). In this project, SVR was used to predict formation energies and potential energies. The following steps were involved:

- **Preprocessing:** Input features were normalized, and outliers were filtered based on properties like energy above the convex hull.
- **Model Tuning:** Bayesian optimization was used to tune hyperparameters such as C, γ, and ϵ.
- **Evaluation:** The model was evaluated using Root Mean Square Error (RMSE), comparing predicted values against the ground truth.

C. Fully Connected Neural Networks (FCNNs)

Fully Connected Neural Networks (FCNNs) are deep learning models where each neuron in one layer is connected to every neuron in the subsequent layer. The FCNN model in this project was designed to predict energy-related properties from DFT features:

- **Architecture:** The network consisted of an input layer, three hidden layers with 128, 64, and 32 units respectively, and an output layer. ReLU activation was used for non-linearity in hidden layers.
- **Training:** The model was trained using the Mean Squared Error (MSE) loss function and the Adam optimizer for 20 epochs. Regular evaluations were performed on test data.
- **Output:** Predicted energy per atom for Si-metal alloy systems.

The FCNN effectively captured complex interactions in the dataset and demonstrated strong predictive performance.

IV. Results

The metric used to compare how good the predictions are was Root Mean Square Error. Root Mean Square Error (RMSE) is a widely used metric for assessing the performance of regression models. It calculates the average magnitude of the error between predicted and actual values, with larger errors being penalized more due to the squaring operation. RMSE is particularly useful for continuous predictions as it retains the same units as the target variable, making it easy to interpret. A lower RMSE value indicates a more accurate model.

The formula for RMSE is given by:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where:

- y_i: Actual value of the i-th data point.
- $\hat{y}_i$: Predicted value of the i-th data point.
- n: Total number of data points.

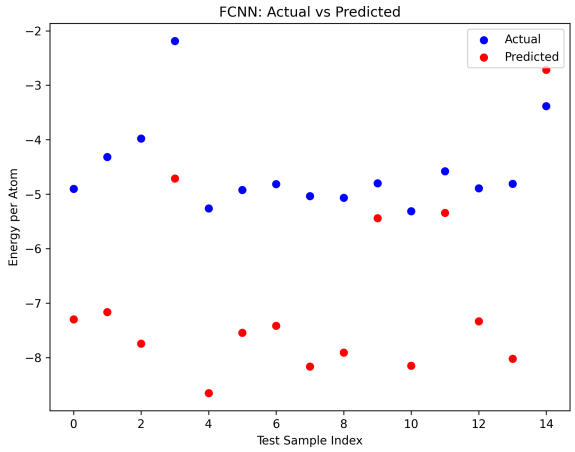

Fig. 1. Actual vs Predicted Potentials by Neural Networks on the X-Ray Diffraction data

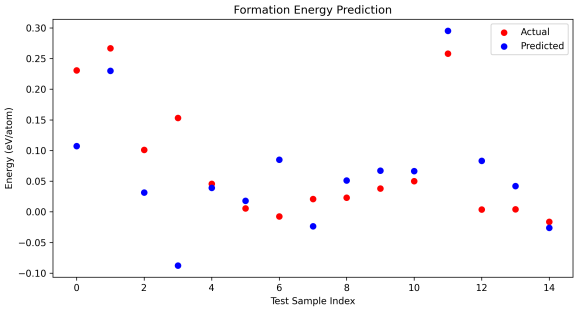

Fig. 2. Actual vs Predicted Potentials by SVR on the X-Ray Diffraction data

A lower RMSE value reflects better model performance by indicating smaller deviations between predictions and actual outcomes.

Dataset	Model	Train RMSE	Test RMSE
XRD	SVR	0.095	0.087
Orbital	SVR	0.126	0.125
Sine	SVR	0.124	0.179
DFT	SVR	0.119	0.131
XRD	GMM	0.476	1.323
Orbital	GMM	0.270	1.364
Sine	GMM	0.543	1.150
DFT	GMM	0.392	1.421
XRD	NN	0.731	1.874
Orbital	NN	1.013	2.057
Sine	NN	1.231	2.012
DFT	NN	1.482	2.391

TABLE I
COMPARISON OF TRAIN AND TEST RMSE

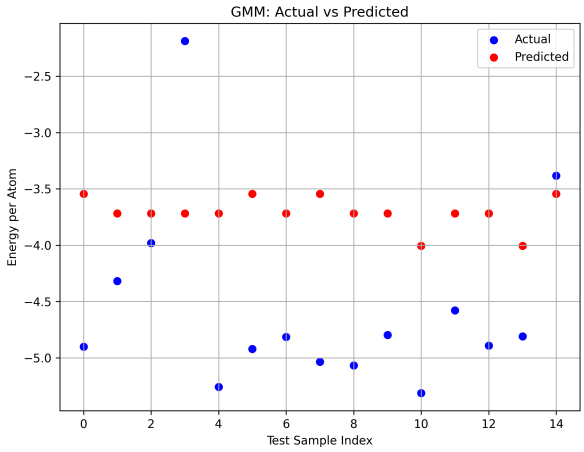

Fig. 3. Actual vs Predicted Potentials by GMMs on the X-Ray Diffraction data

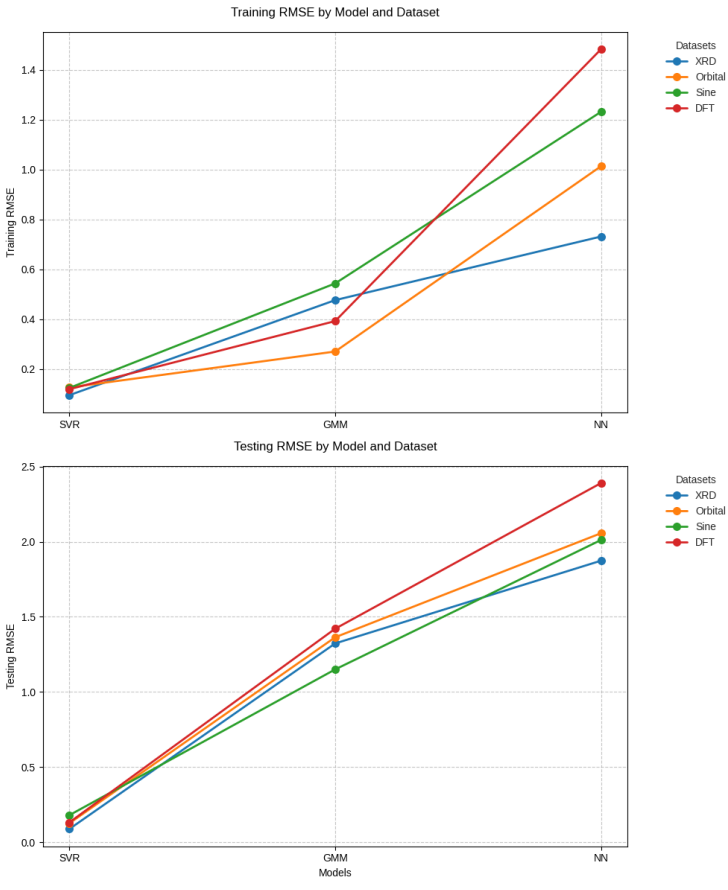

Fig. 4. RMSE Variation for different models using both test and train data

V. Inferences from Results

Training RMSE

- **SVR Model:** Achieves the lowest training RMSE across all datasets, ranging from 0.095 to 0.126. This indicates SVR's strong capability to fit training data effectively.
- **GMM Model:** Exhibits moderate training RMSE values, with a range of 0.270 to 0.543, showing it is less effective than SVR but generally better than NN on training data.
- **NN Model:** Records the highest training RMSE values across datasets, ranging from 0.731 to 1.482, which suggests overfitting or difficulty in fitting the training data.

Testing RMSE

- **SVR Model:** Maintains relatively low testing RMSE values (0.087 to 0.179) across all datasets, indicating good generalization performance.
- **GMM Model:** Testing RMSE values (1.150 to 1.421) are significantly higher than training RMSE, suggesting potential overfitting or limited generalization ability.
- **NN Model:** Testing RMSE values are the highest among the models (1.874 to 2.391), reflecting poor generalization and overfitting to the training data.

Dataset-Specific Observations

- **XRD Dataset:** All models perform relatively well, with SVR showing the best performance in both training and testing RMSE.
- **Orbital Dataset:** Similar trends are observed as in the XRD dataset, with SVR outperforming the other models.
- **Sine Dataset:** SVR shows slightly higher testing RMSE (0.179), but still performs better than GMM and NN.
- **DFT Dataset:** All models struggle, with NN performing the worst. SVR still demonstrates better relative performance.

Conclusions

- SVR consistently outperforms GMM and NN in both training and testing RMSE across all datasets, highlighting its robustness.
- NN exhibits poor performance, particularly on testing data, suggesting overfitting and a need for better regularization or architecture tuning.
- GMM shows moderate performance but struggles with generalization, indicating potential limitations in its modeling assumptions.

VI. Future Work

Model Improvements and Optimization

Future work should focus on improving the performance of the existing models:

- **SVR:** Fine-tuning hyperparameters such as the kernel function and regularization parameters may lead to better model performance.
- **GMM:** Optimizing the number of components and covariance types could improve model generalization.
- **NN:** Neural networks showed poor generalization, and exploring advanced regularization techniques (e.g., dropout, L2 regularization) and tuning the architecture could mitigate overfitting.

Ensemble Learning

Combining multiple models using ensemble learning methods could help improve overall performance:

- Methods such as *boosting*, *bagging*, or *stacking* could be applied to reduce bias and variance across models.
- A hybrid approach, integrating SVR for precision and NN for learning complex patterns, might result in superior predictive accuracy.

Code

All codes used in this project are at **this link.** The `readme.md` file has instructions to replicate the results.

Acknowledgment

I thank Professor Amit Singh for teaching us in the semester and giving me the tools to complete this project. His demonstrations in class were very insightful and enabled me to complete this fruitful project.

References

- Ward, L., Dunn, A., Faghaninia, A., Zimmermann, N. E. R., Bajaj, S., Wang, Q., Montoya, J. H., Chen, J., Bystrom, K., Dylla, M., Chard, K., Asta, M., Persson, K., Snyder, G. J., Foster, I., Jain, A., Matminer: An open source toolkit for materials data mining. Comput. Mater. Sci. 152, 60-69 (2018).
- Sun, Qiming, et al. "PySCF: the Python-based simulations of chemistry framework." Wiley Interdisciplinary Reviews: Computational Molecular Science 8.1 (2018): e1340.
- Hart, G.L.W., Mueller, T., Toher, C. et al. Machine learning for alloys. Nat Rev Mater 6, 730–755 (201). https://doi.org/10.1038/s41578-021-00340-w