Machine learning assisted prediction of thermal conductivity

Final Project Submission

Submitted by: Team 2

Shahid Ahmed, 22M1626

Mithun M Nair, 22M1628

Praveen Kumar Yadav, 22M1640

Introduction to the problem

Motivation

Predict thermal conductivities for high throughput material screening in contrast to highly resource intensive traditional methods like Lattice Dynamics and Molecular Dynamics.

Why Metal Oxides?

- Excellent waste heat recovery in industrial power plants
- High electric conductivity and Seebeck coefficient can be achieved by transient element doping and band gap engineering.

Lattice Dynamics and Molecular Dynamics

❑Lattice Dynamics :
Thermal conductivity is predicted from first principles by studying phonon characteristics.

❑Molecular Dynamics :
Thermal conductivity is predicted from a classical approach by analyzing the atomic trajectories using Green Kubo autocorrelation function at equilibrium or Fourier's law of heat conduction at a non equilibrium steady state

Utility of Machine Learning

❑Pseudopotentials :
Pseudopotentials can be developed with the help of Machine Learning algorithms to be used with LD and MD.

❑Correlating features to the Thermal conductivity:
Identifying relevant features from reading papers and using a database to extract them. Correlating the features to the thermal conductivity with the help of Neural Networks.

Data Collection

Steps involved

- Extensively reviewed literature to get an idea of properties correlated with thermal conductivity.
- AFLOW database was selected to obtain different properties for the oxides.
- Matminer was used to extract data from AFLOW database and convert it into a data frame. Compounds containing oxygen “O” were selected by setting {“species”: “O”} as the criteria, while the required properties of the compound were specified as a list argument.
- The Keywords file from AFLOW API package was modified to extract desired properties by the addition of appropriate classes.
- The obtained dataset was cleaned and standardized using python libraries.

Oxides	Lattice_type	Bulk_Modulus	Density	Heat_capacity	Debye_Temperature	Gruneisen_parameter	Egap	Poisson_ratio	Thermal_expansion	Unitcell_atom	Speed_of_soun	Thermal_Conductivity
Ag4O2	CUB	50.0423	6.62411	19.566	56.0669	2.48656	0.0868	0.487902	0.00011511	6	498.802	0.0174529
Cu4O2	CUB	97.0286	6.01944	18.7031	155.957	2.25972	0.6558	0.466027	7.51E-05	6	1237.21	0.253177
O4Pt2	CUB	134.219	8.4695	18.4381	136.007	2.31201	0	0.470661	4.91E-05	6	1089.3	0.262206
O2Pd4	CUB	117.208	8.1206	18.5642	144.925	2.44995	0	0.462663	5.72E-05	6	1187.39	0.287144
Cl16Hg12O4	CUB	39.5693	5.77917	101.107	181.028	2.37251	1.4766	0.336375	9.62E-05	32	994.214	0.294549
Bi4Cs12O12	CUB	32.2056	5.17338	88.2131	164.057	2.16724	3.0558	0.340246	9.69E-05	28	1143.55	0.296848
Cs12O12Sb4	CUB	35.5356	4.59582	87.6459	182.911	2.20131	3.0799	0.341438	9.07E-05	28	1257.23	0.343002
Ag5O6Pb2	HEX	82.1727	8.32514	40.4582	193.109	2.40299	0	0.404837	7.78E-05	13	1469.84	0.452188
O24Ti16	BCC	105.162	9.72744	121.641	244.123	2.42234	0	0.364602	6.19E-05	40	1985.77	0.474624
Cs6Ni2O4	TET	27.3577	4.68948	38.5288	171.929	2.23058	1.3118	0.32228	0.00012407	12	1563.44	0.488223
Cd4O8Rb8	ORC	40.105	4.70755	62.911	215.957	2.21737	1.3563	0.334848	0.000107384	20	1864.6	0.492799
Bi16O24	BCC	110.209	8.74592	120.166	223.069	2.09739	1.8636	0.391946	4.46E-05	40	1875.06	0.526131
Ag4O4Rb4	BCT	30.841	4.47728	38.6945	197.895	2.35511	1.4461	0.314963	0.000130678	12	1646.41	0.545209
Bi4O12Rb12	CUB	36.8712	4.72483	87.0688	224.034	2.15065	3.1139	0.28736	9.63E-05	28	1409.36	0.572559
Au4O4Rb4	BCT	40.5837	6.51611	38.0423	191.987	2.405	1.683	0.309732	0.000102381	12	1431.19	0.672033

Fig 1: A part of dataset where the column headers show the properties selected after extensively reviewing literature. The dataset is standardized and a principal component analysis is done.

Data Representation

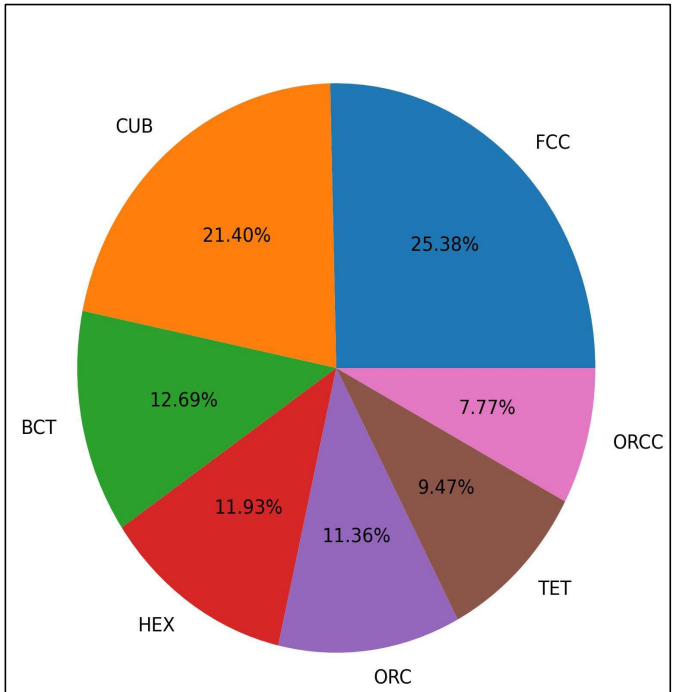


Fig 2: The pie chart shows the distribution of the oxides into different Bravais Lattices. Monoclinic and Body centred Orthorhombic has a very small population thus such oxides can be dropped for further studies

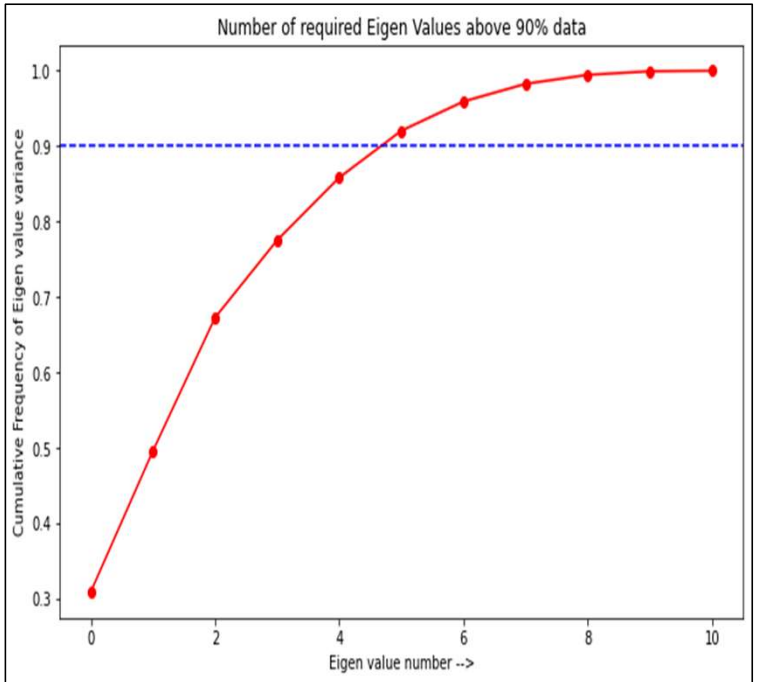


Fig 3: Heat Map denotes the correlation of the PCs with the thermal conductivity in the last column. Each PC is correlated with each other at the order of 10^{-15} which clearly shows that there is no data redundancy.

Data Representation

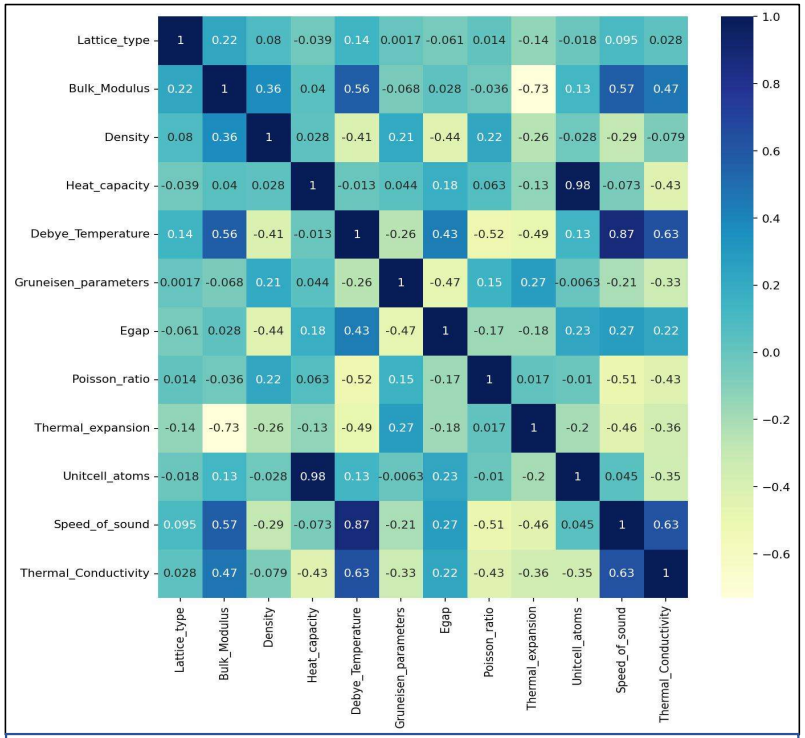


Fig 4: Heat Map denotes the correlation of the different parameters with each other. The last column denotes the correlation of each property with the thermal conductivity. Here we have used a Pearson correlation.

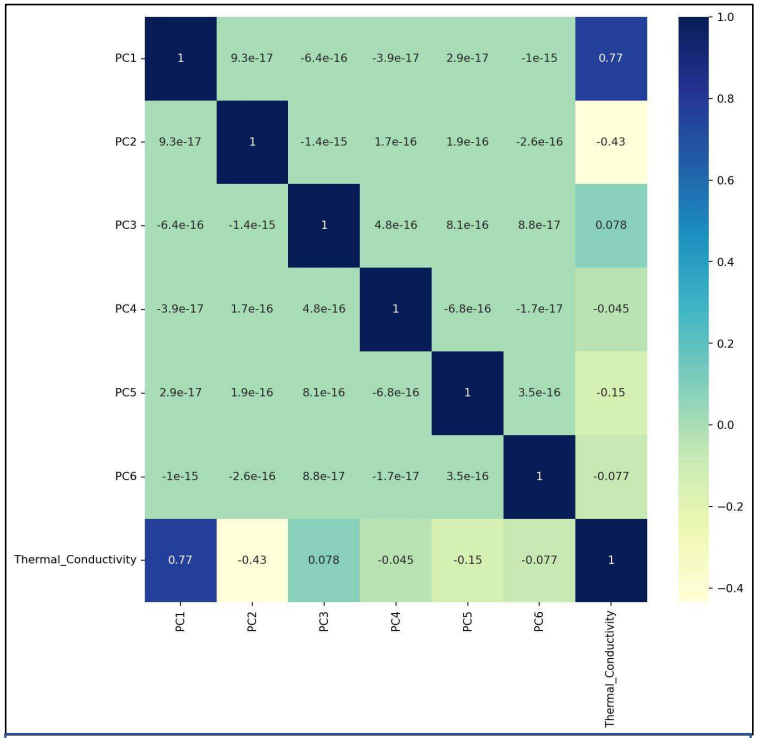
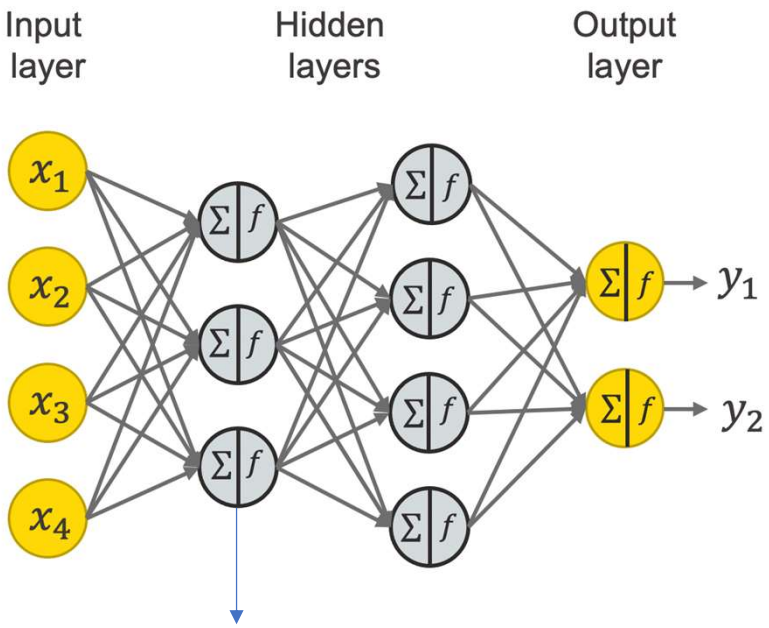


Fig 5: Heat Map denotes the correlation of the PCs with the thermal conductivity in the last column. Each PC is correlated with each other at the order of 10^{-15} which clearly shows that there is no data redundancy.

Artificial Neural Networks



A Neuron consisting of:

- Linear function
- Non-linear activation function

Description

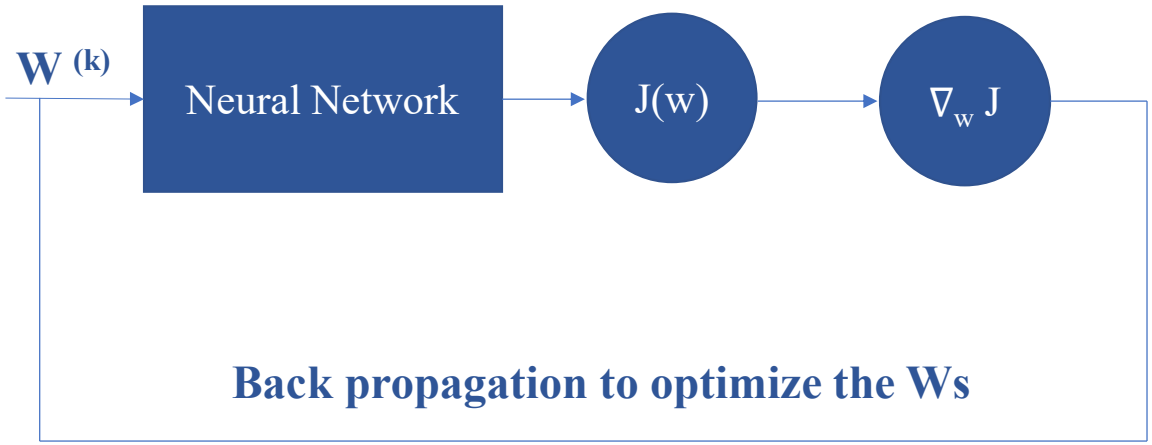
ANN consists of different layers like the input, hidden and the output layers. These layers have various neurons which are connected with the neurons of the adjacent layers.

The neurons in a layer consists of two functions:

- A linear function which basically takes the linear combination of the activation from the previous layer
- A non linear function which operates on the output of the linear function to give the output/activation of the neuron.

Image source: *https://www.knime.com/blog/a-friendly-introduction-to-deep-neural-networks*

Optimization of an ANN



Process

- Start with a guess for W [weights & biases]
- Run through the NN and find the value of J(w) [Loss function]
- Find $\nabla_w J$ [Grad of the loss function with respect to the weights] using back prop algorithm using an optimizer
- Ws are updated based on the $\nabla_w J$ until an optimized model is achieved where $\nabla_w J = 0$ or goes below a stopping criteria and the iterations stops else we proceed to next step.

Optimizers

Momentum Method

$$\delta w_t = \gamma . \delta w_{t-1} + \alpha . g_{t-1}$$

$$w_{t+1} = w_t - \delta w_t$$

RMSprop

$$G_{i,t} = \beta * G_{i,t-1} + (1 - \beta) * (g_{i,t})^2$$

$$w_{i,t+1} = w_{i,t} + \frac{\alpha}{\sqrt{G_{i,t} + \epsilon}} g_{i,t}$$

$$g_t = \nabla_w J(w_t) \qquad g_{i,t} = \frac{\partial J}{\partial w_{i,t}}$$

ADAM

$$m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * g_t$$

$$v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * (g_t)^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \qquad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

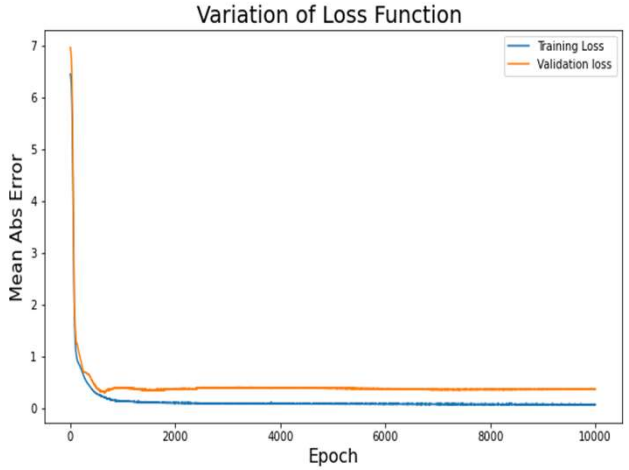
$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t$$

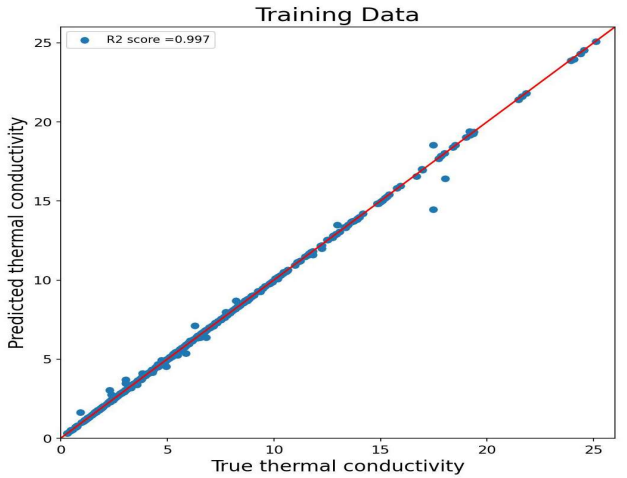
Adam optimizer is based on momentum and RMS prop

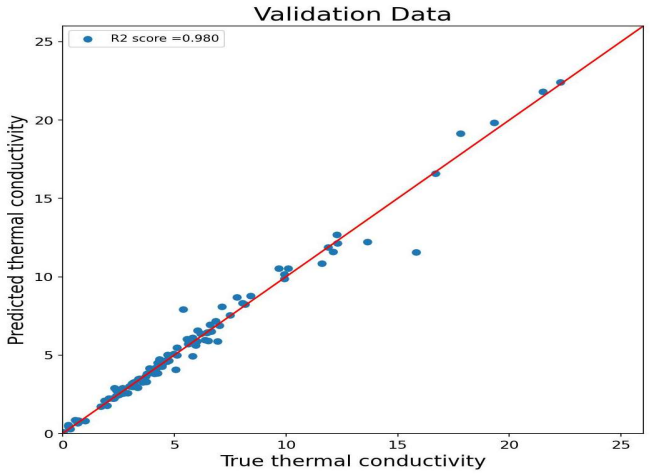
Feed Forward DNN with Complete Dataset

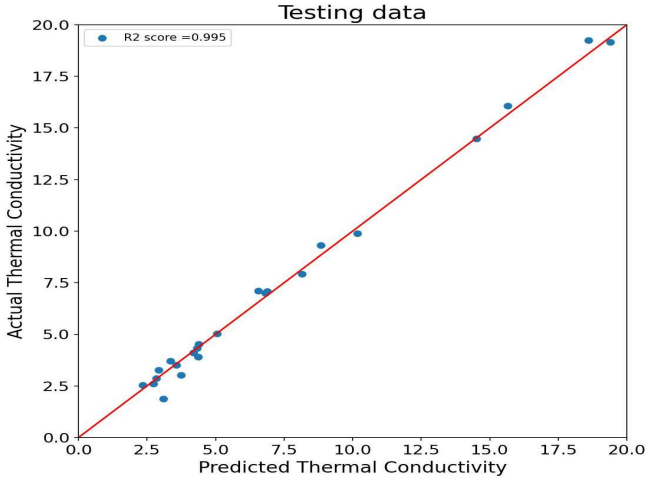
Parameters used for results:

- No. of layers: 4
- Layer1 = 16 neurons
- Layer2 = 32 neurons
- Layer3 = 128 neurons
- Layer4 = 1 neuron
- Activation function = “relu”
- EPOCHS = 10000
- Learning rate = 0.001
- R2_score (Testing Data) = 0.995
- Total datapoints = 524
- Training datapoints = 400
- Validation datapoints = 100
- Testing datapoints = 24
- Optimizer = **RMSprop**





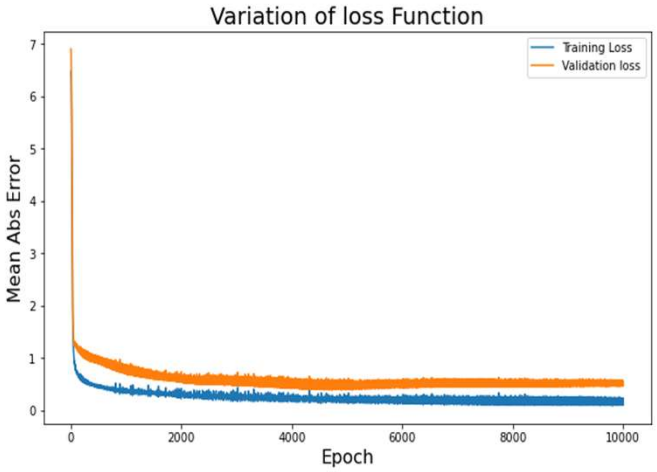


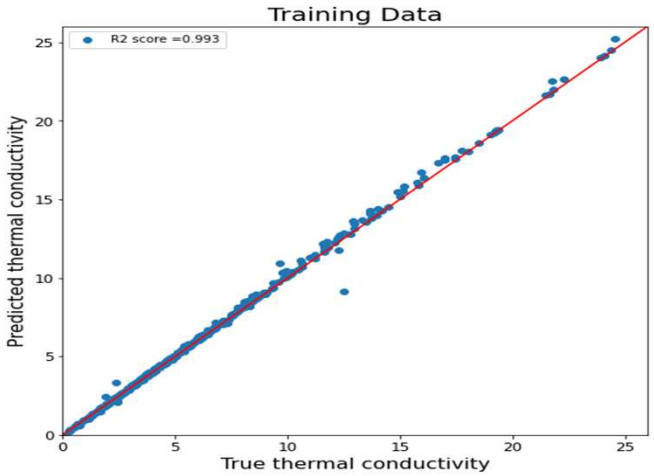


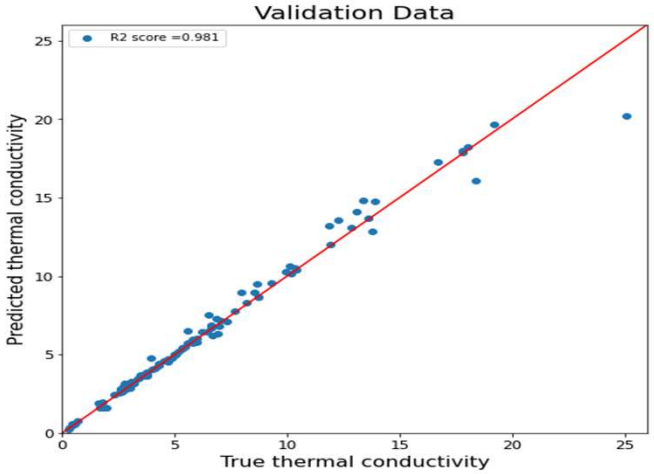
Feed Forward DNN with Complete Dataset

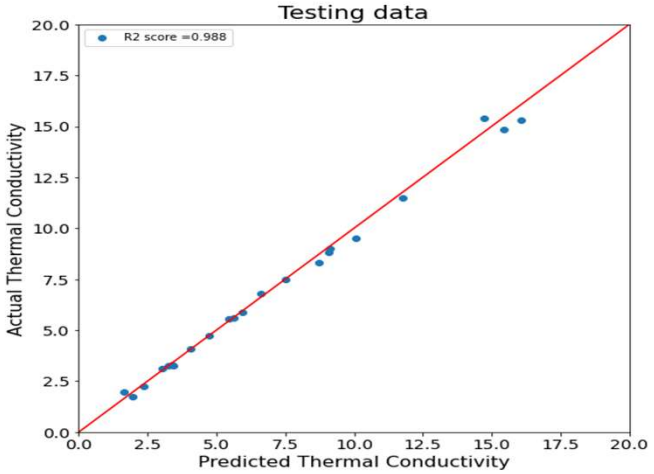
Parameters used for results:

- No. of layers: 4
- Layer1 = 16 neurons
- Layer2 = 32 neurons
- Layer3 = 128 neurons
- Layer4 = 1 neuron
- Activation function = “relu”
- EPOCHS = 10000
- Learning rate = 0.001
- R2_score (Testing Data) = 0.988
- Total datapoints = 524
- Training datapoints = 400
- Validation datapoints = 100
- Testing datapoints = 24
- Optimizer = **Adam**





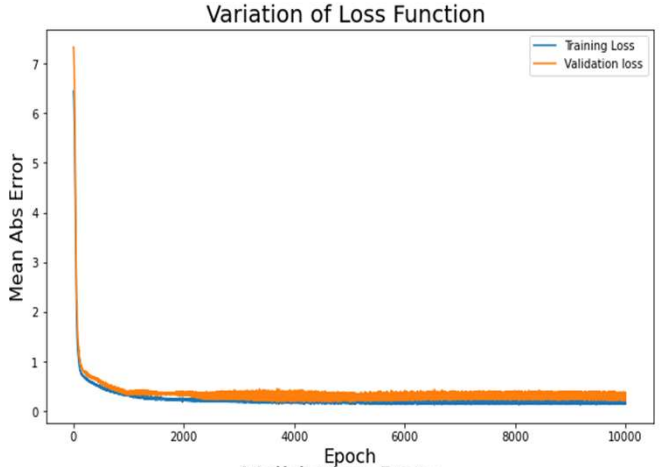


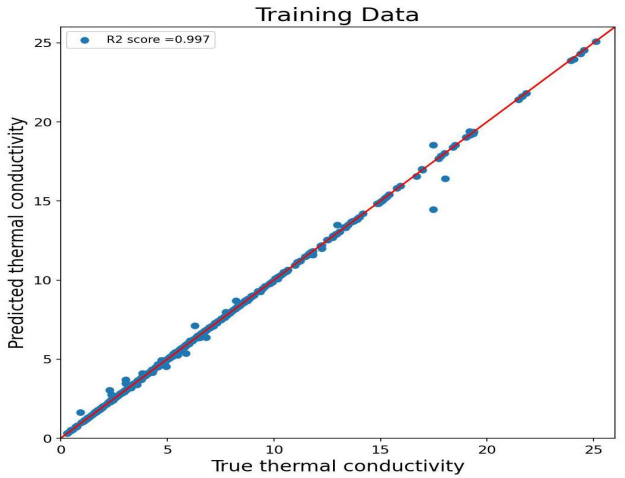


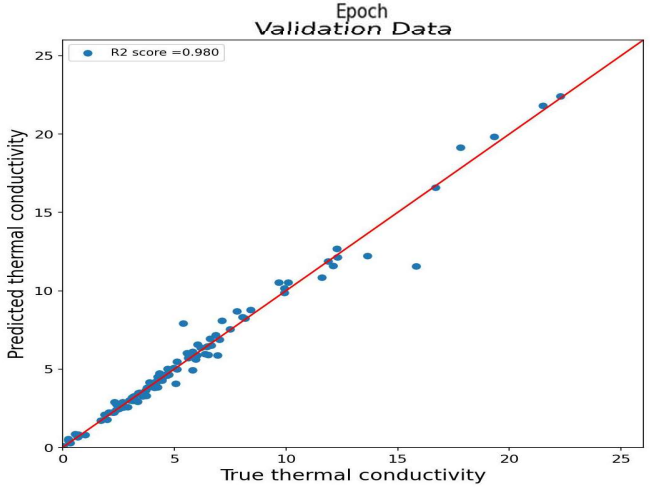
Feed Forward DNN with PCAs

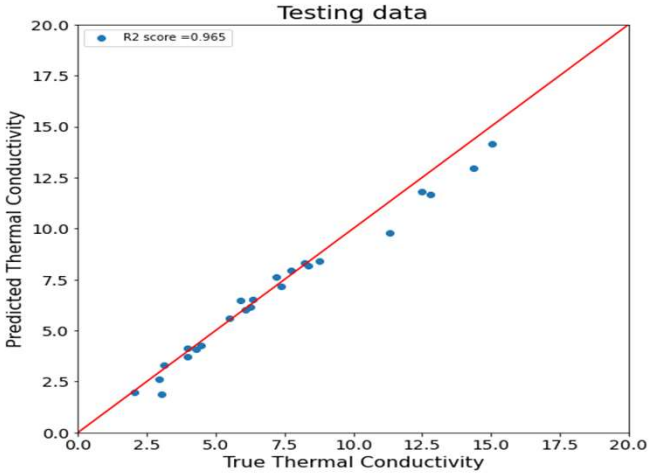
Parameters used for results:

- No. of layers: 4
- Layer1 = 16 neurons
- Layer2 = 32 neurons
- Layer3 = 128 neurons
- Layer4 = 1 neuron
- Activation function = “relu”
- EPOCHS = 10000
- Learning rate = 0.001
- R2_score (Testing Data) = 0.965
- Total datapoints = 524
- Training datapoints = 400
- Validation datapoints = 100
- Testing datapoints = 24
- Optimizer = **RMSprop**









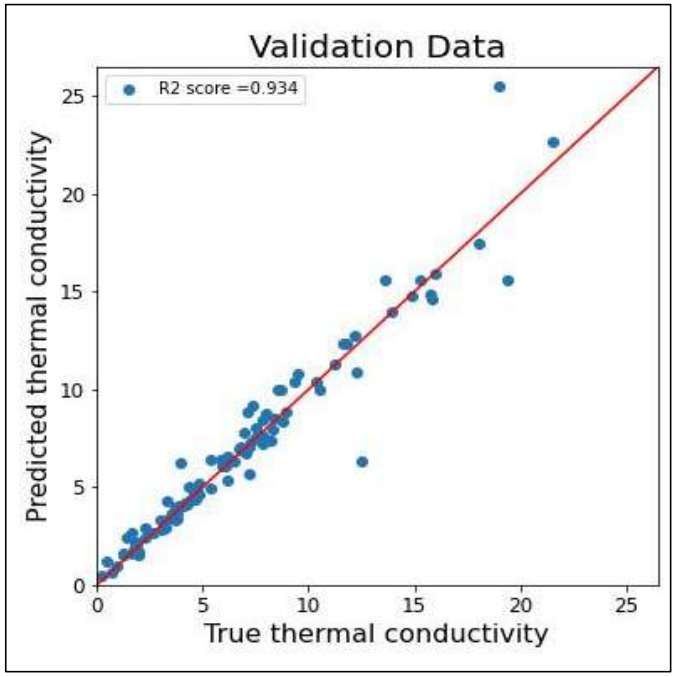
Feature study

Features	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21
Density	1	0	0	0	0	0	0	0	0	0	0
Heat Capacity	0	1	0	0	0	0	0	1	1	1	1
Thermal Expansion	0	0	1	0	0	0	0	1	1	1	1
Speed of sound	0	0	0	1	0	0	0	1	1	1	1
Bulk Modulus	0	0	0	0	1	0	0	0	1	0	1
Poisson Ratio	0	0	0	0	0	1	0	0	0	0	0
natoms	0	0	0	0	0	0	1	0	0	1	1
						R2 Score					
Train	0.002	0.254	0.307	0.408	0.168	0.19	0.095	0.911	0.946	0.938	0.944
Valid	-0.003	0.27	0.148	0.343	0.209	0.215	0.125	0.894	0.841	0.912	0.89
Test	-0.057	0.442	0.438	0.221	0.29	-0.18	0.236	0.903	0.94	0.946	0.978
Train	-0.094	0.237	0.247	0.398	0.19	0.229	0.125	0.921	0.938	0.951	0.937
Valid	-0.124	0.266	0.25	0.369	0.182	0.126	0.115	0.824	0.816	0.882	0.893
Test	-0.113	0.521	0.402	0.169	0.281	-0.208	0.295	0.917	0.883	0.943	0.939

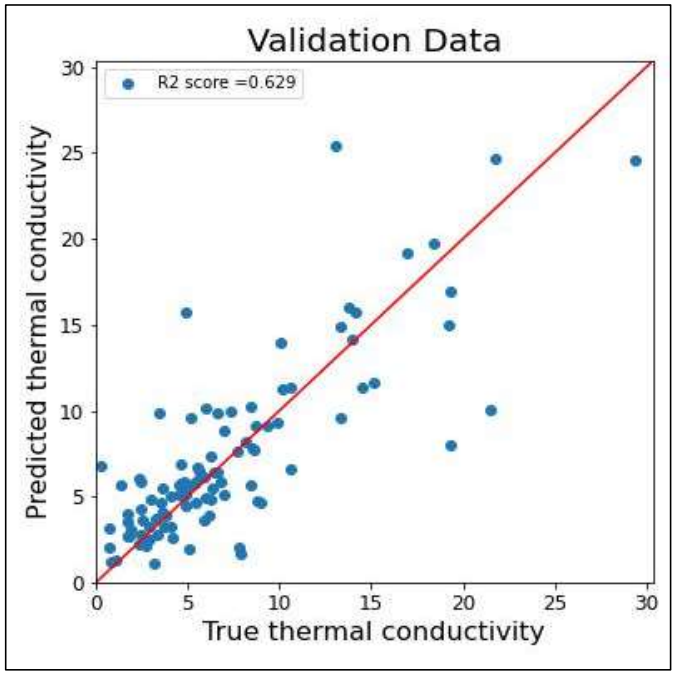
- Variance captured by individual features were studied.
- Features with maximum captured variance were used together and a much accurate model is obtained.

Number of features: A comparative study

Features	Case 1	Case 2
Unitcell_atoms	✓	✓
Lattice_type	✓	✓
Bulk_Modulus	✓	✓
Density	✓	✓
Poisson_ratio	✓	×
Thermal_expansion	✓	×
Heat_capacity	✓	×
Speed_of_sound	×	×
Debye_Temperature	×	×
Gruneisen_parameters	×	×
Egap	×	×

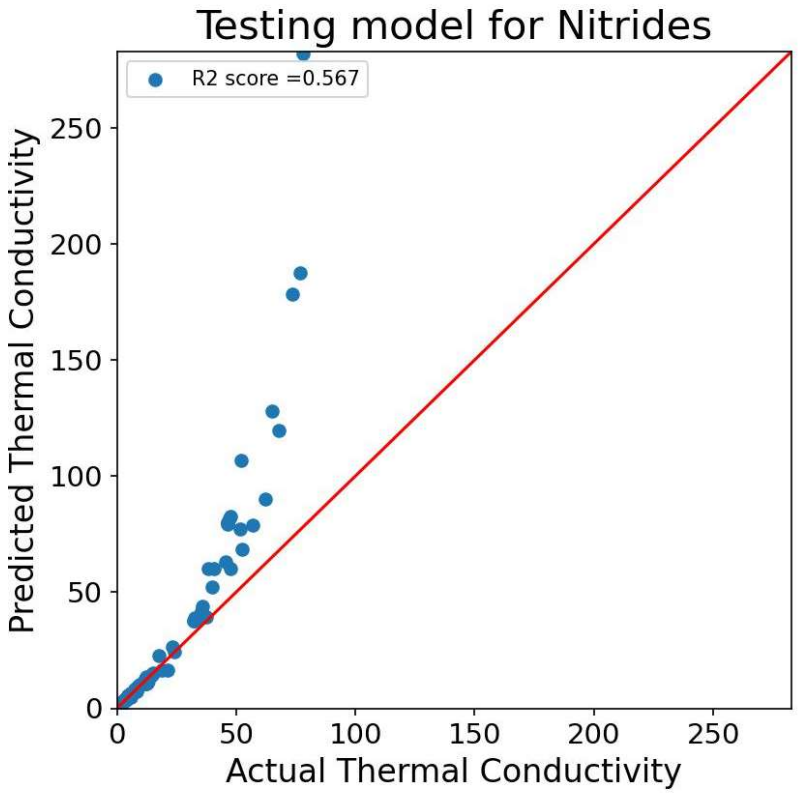


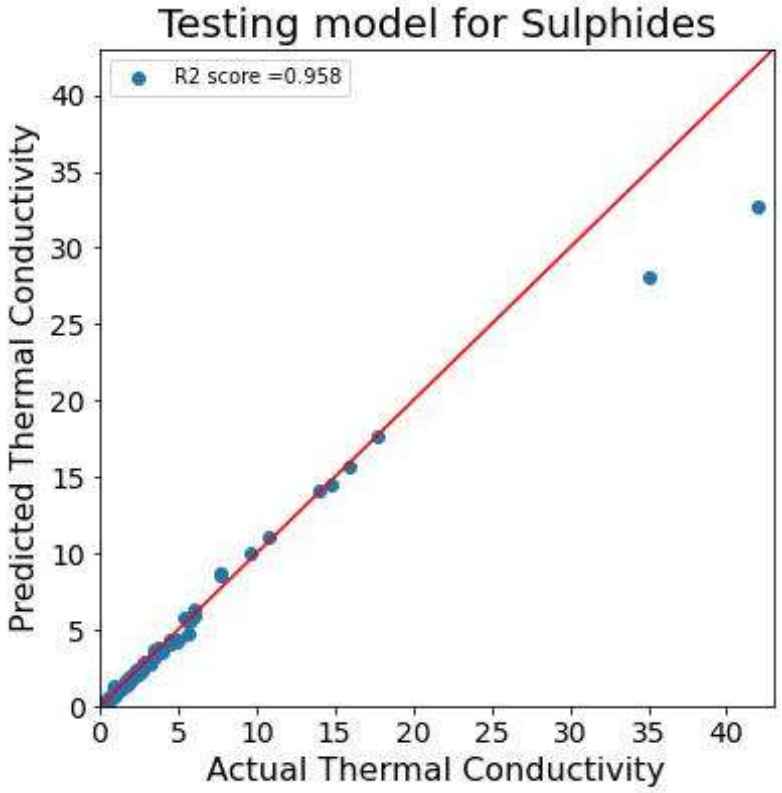
Case 1



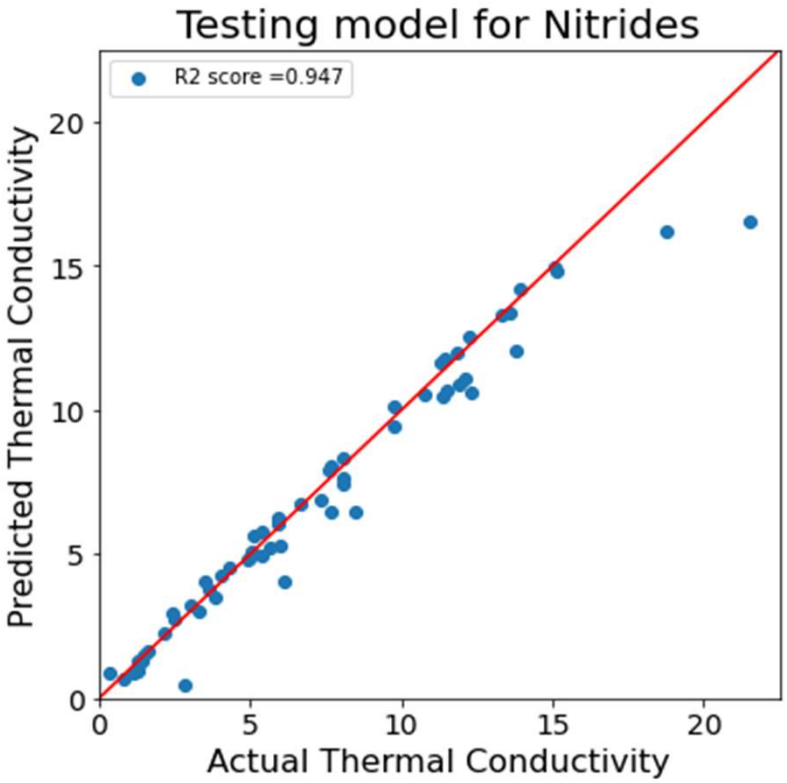
Case 2

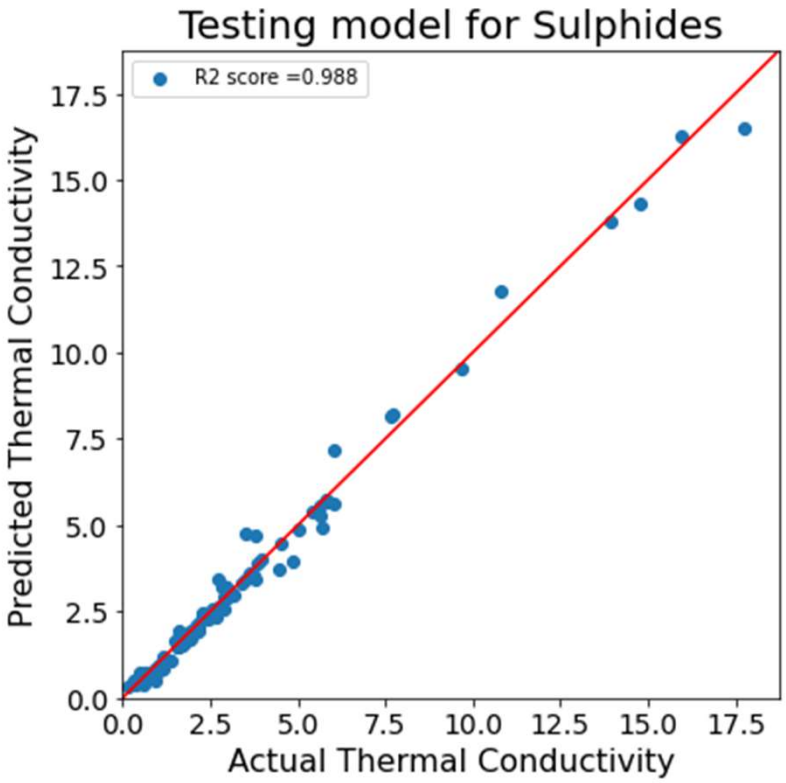
Extension of study





Extension of study





Discussion

- Derived parameters such as Debye temperature and Gruneisen parameters were dropped because they are obtained using computationally expensive calculations such as DFT. Dropping these features did not have any significant drop in the R2 score.
- Basic features were studied to understand their contribution to thermal conductivity (κ). It was found that heat capacity, thermal expansion and speed of sound captured the most variance and their combination provided an accurate κ prediction.

$$\kappa = \frac{1}{V}\sum_{\lambda} c_\lambda v_\lambda^2 \tau_\lambda$$

- Dropping these features leads to a significant drop in the R2 score and the model can no longer predict the thermal conductivity values accurately.

Conclusion

- Thermal conductivity values were predicted with high accuracy for oxides with low thermal conductivity.
- Using the PCA space with 6 features, the model has a high R2 score of 0.96 compared to the 0.99 R2 score obtained using 11 features with the RMSprop optimizer.
- The model requires more training with higher thermal conductivity oxides so that it can predict the thermal conductivities more efficiently.
- The model can also accurately predict the thermal conductivity of sulphides. This could be due to an equal number of valence electrons.
- The model can also predict the thermal conductivities of nitrides accurately for low thermal conductivity nitrides.

References

- Tewari, A., Dixit, S., Sahni, N., & Bordas, S. P. A. (2020). Machine learning approaches to identify and design low thermal conductivity oxides for thermoelectric applications. *Data-Centric Engineering*, *1*(6).
- Rose, F., Toher, C., Gossett, E., Oses, C., Nardelli, M. B., Fornari, M., & Curtarolo, S. (2017). AFLUX: The LUX materials search API for the AFLOW data repositories. *Computational Materials Science*, *137*, 362–370.
- 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).

Link to codes: *https://drive.google.com/drive/folders/18eEh9KpIfn_GLMT096WBNHA6d_C2ASv1?usp=sharing*