Jadavpur University, Kolkata

Internship Report

Name: Soumyadeep Dey
Department: Mechanical Engineering
College: National Institute of Technology, Durgapur
Internship Title: Hybrid CFD-ML Approach To Predict 3D Fin Temperature Distributions Using 2D Fin Mid-Plane Data
Duration: 2nd June, 2025 – 4th July, 2025
Organization: Mechanical Engineering Department, Jadavpur University, Kolkata
Supervisor (Mentor): Dr. Achintya Mukhopadhyay

1. Abstract

Recent developments in machine learning and AI field have opened new doors for efficient prediction of heat transfer characteristics in complex geometries. Inspired from the studies, this work proposes an Artificial Neural Network (ANN) based approach to predict 3D temperature fields using limited 2D thermal data. The model is trained on the temperature difference (ΔT) between a central 2D mid-plane data at z = 0 and various 3D planes along the z-axis. The difference is computed as $\Delta T(x, y, z) = T_{3D(x,y,z)} - T_{2D(x,y,z=0)}$ and the ANN learns this ΔT distribution across different z-planes with different x and y coordinates. During prediction, the model outputs ΔT for desired z-planes, which are then superimposed on the known 2D temperature plane to reconstruct full 3D temperature fields. This approach reduces the need for dense sensor data while maintaining high spatial accuracy, and is validated using datasets inspired by recent literature on micro-pin fin heat sinks and multimodal ML applications in heat transfer. The proposed model demonstrates the potential of physics guided machine learning for efficient 3D thermal field prediction.

2. Introduction

Heat transfer is quiet an essential task due to the integration of I.C. chips in every field. It can be physically possible by using fins or extended surfaces. Computing heat transfer using 3d find is an expensive task so we use 2d fins to study the characteristics, but in 2d fins the results are being compromised, hence we developed a machine learning approach to bridge the gap between the 2d and 3d simulations without compromising the data.

In recent years, machine learning (ML) techniques, particularly Artificial Neural Networks (ANNs), have emerged as powerful tools for modelling and predicting thermal behaviour based on limited input data. Several studies have demonstrated the ability of ANNs to capture nonlinear spatial patterns and predict temperature fields with high accuracy. Notably, prior research has applied ML to micro-pin fin heat sinks and other thermal systems, showing promising results in reducing computational cost while maintaining prediction accuracy.

Building on this foundation, the present work proposes an ANN-based model that predicts the 3D temperature field using a known 2D temperature plane at z = 0. The model is trained to learn the temperature difference, $\Delta T(x, y, z) = T_{3D(x,y,z)} - T_{2D(x,y,z=0)}$ allowing it to infer the 3D thermal distribution across multiple z-planes. By adding the predicted ΔT to the known 2D temperature data along the same x and y coordinates, the full 3D temperature field can be reconstructed. This approach significantly reduces the dependency on dense 3D sensor data and expensive simulations, offering a fast and scalable alternative for thermal analysis in engineering applications.

Nomenclature

w_i	Weight matrix for layer i (dimensionless)
b_i	Bias vector for layer i (dimensionless)
x_i	Input value for the model (dimensionless)
y_i	Predicted temperature for the i^{th} sample (K)
y_i	Ground truth temperature for the i^{th} sample (K)
f(.)	Activation function (dimensionless)
MSE	Mean Square Error (dimensionless)
MAE	Mean absolute error (dimensionless)
O_i	Output of neuron i (dimensionless)

Greek Symbols

η	Learning rate for Adam optimizer
δ	Error term at neuron j

Subscripts

src	Source
tgt	Target
i	Index for data sample or network layer

2.1 Literature Review

Micro-pin fin heat sinks (MPFHSs) are widely used for cooling high heat flux electronic systems due to their large surface area and low pressure drops. Studies have explored how geometry, spacing, and flow conditions affect their performance, leading to various empirical models, though these often lack flexibility across different setups. Machine learning (ML) models like ANN, XGBoost, and LightGBM have shown better accuracy with lower error rates. Kim et al.

used 906 data points from 15 studies to develop ML models with 7.5–10.9% MAE, outperforming traditional methods. **[1]**

With the rise of 3D integrated circuits (3D-ICs), thermal management has become crucial due to increased heat from stacking. MPFHSs are suitable for cooling 3D-ICs, but predicting boiling heat transfer remains challenging. ML models have shown better performance than traditional approaches, especially with two-phase flow. Multimodal ML models using both numerical and image data (e.g., boiling patterns) have achieved high accuracy—MAPE of 0.84% for average and 1.81% for maximum temperature—highlighting their potential for improved thermal design. **[2]**

Natural convection cooling with vertical fins on horizontal surfaces has been widely studied for passive systems. Performance depends on fin number, spacing, length, shape, and material. General Nusselt number correlations (Ra, Pr) were proposed by Churchill and Chu. Numerical studies confirm heat transfer increases with fin count until overcrowding reduces performance, with an optimal S/L ratio around 0.08421. **[3]**

3. Objectives of the internship

I. Develop a Machine Learning Model:
   To design and implement an Artificial Neural Network (ANN) capable of learning temperature difference patterns between 2D and 3D thermal data across different x and y coordinates for different z planes.

II. Predict 3D Temperature Fields Using 2D Data:
   To accurately reconstruct 3D temperature distributions from a known 2D central plane data by predicting ΔT values across various z-planes.

III. Reduce Computational Dependency:
   To minimize reliance on full-scale 3D simulations or dense sensor data by creating a lightweight and efficient predictive model.

IV. Validate the Model with Real/Simulated Data:
   To evaluate the ANN model's accuracy and generalization ability using simulated datasets weather it is physically possible or not.

4. Tools and Technologies Used

During the internship, I gained practical hands-on experience on the following tools and technologies:

- Python:
  Used as the programming language for data preprocessing, analysis, visualization, and model development. Libraries used are NumPy, Pandas, Matplotlib, and Scikit-learn (for data handling and splitting).

- Artificial Neural Networks (ANN):
  The primary machine learning approach used for modelling and prediction. The ANN was built using Dense layers with Relu activation functions in hidden layers, and a Linear activation function in the output layer. The model was trained using the Adam optimizer over multiple epochs, with Early Stopping applied based on validation loss to avoid overfitting. Both training and validation loss curves were analysed to monitor performance.

- ANSYS Workbench (Basic Use):
  Gained introductory experience with finite element simulations for heat transfer analysis across geometries of extended fin surfaces.

- Microsoft Excel & Word:
  Used for quick data cleaning, tabulation, and reporting of experimental and model results.

- Jupyter Notebook:
  Served as the main development environment for coding and simulation, model experimentation, and visualization.

```
model = Sequential([
    Dense(64, activation = 'relu', input_shape = (3,)),
    Dense(128, activation = 'relu'),
    Dense(64, activation = 'relu'),
    Dense(32, activation = 'relu'),
    BatchNormalization(),
    Dense(1, activation = 'linear')
])

model.compile(optimizer = 'adam', loss = 'mse', metrics = ['mae'])
early_stop = EarlyStopping(monitor = 'val_loss', patience = 5, restore_best_weights = True)
```

```
history = model.fit(
    X_train, y_train,
    validation_data = (X_test, y_test),
    epochs = 20,
    batch_size = 32,
    callbacks = [early_stop],
    verbose = 1
)
```

Fig 1. The code snippet of the ANN model build

5. Methodology and Workflow

The following workflow has been implemented throughout the internship:

5.1 Learning About the Work and Past Researches.

- I began by reading several research papers related to machine learning applications in heat transfer and fin simulations. These provided a strong foundation of the idea.
- Throughout the initial weeks, I regularly interacted with mentors and Ph.D. students to gain insights into how to structure the project and choose the right modelling approach.
- Multiple handwritten drafts, model diagrams, and strategies were created to finalize the direction of the project and design the prototype.

5.2 Understanding the Tools and Software

- I installed and explored ANSYS Workbench to simulate basic thermal models in both 2D and 3D geometries.
- A simple cube model was created to simulate temperature flow, and the simulation output was exported in .txt format for further use.
- A basic machine learning model was developed to test the feasibility of our prototype.
- A final workflow pipeline was established, defining which simulation data would be used for training and which would be reserved for prediction.

5.3 Data Extraction and Preprocessing

- In actual scenario, we have developed 2 fin structure model in ANSYS to get the temperature data.
- The data is saved in a .txt file format.
- The simulation output in .txt format was cleaned and converted into structured .csv files.
- Data preprocessing steps included filtering, handling missing values, and organizing data into training and test sets.

5.4 Model Development

- At first, x (input data) and y (output data) were scaled using Standard Scaler to have mean = 0 and variance = 1, where input data corresponds to x and y coordinates along with different z planes and output data corresponds to the ΔT.

- An Artificial Neural Network (ANN) model was built to predict temperature differences (ΔT) across z-planes for different x and y coordinates.
- Model predictions were compared with ANSYS simulation results to test the accuracy.
- Visualization tools such as scatter plots were used for visual comparison of actual and predicted temperature fields.
- Error analysis was performed to identify regions where the model underperformed.

5.5 Validation and Visualization

- Final model outputs were validated using unseen data.
- The 3D temperature fields were reconstructed by combining predicted ΔT with the known 2D input plane.
- Scattered plots were created for the predicted data and the actual data side by side.

5.6 Equations Used:

Mean absolute error formula

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y_i}| \qquad (1)$$

Mean square error formula

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y_i})^2 \qquad (2)$$

Deep learning model equation is given as

$$y = \sum_{i=1}^{n} w_i x_i + b \qquad (3)$$

ReLU activation function equation is

$$\text{ReLU}(x) = \max(0, x) \qquad (4)$$

Piece-wise Form:

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

Standard Scaler for scaling operations

$$x_{scaled} = \frac{x - \mu}{\sigma} \qquad (5)$$

6. Results and Analysis

The integration of ANN has given a generalized and predictive output. ANN models yielded MAE score of 0.35 for over all the epochs.

Key observations: - ML models accurately learned and generalized the temperature pattern in symmetric and linear heat flow regions. To a little extent ANN models were able to approximate complex curved thermal profiles especially in regular fin shapes. Scatter plots have shown the variation well accurately with a maximum temperature difference of 12.54 for z = 100 mm plane and 24.453 for z = 375 mm plane. Comparison between ANSYS simulation and ML predictions showed that with a trained model, prediction time was reduced from 20 minutes to under 1 minute with decent accuracy, highlighting the speed and efficiency of AI methods.

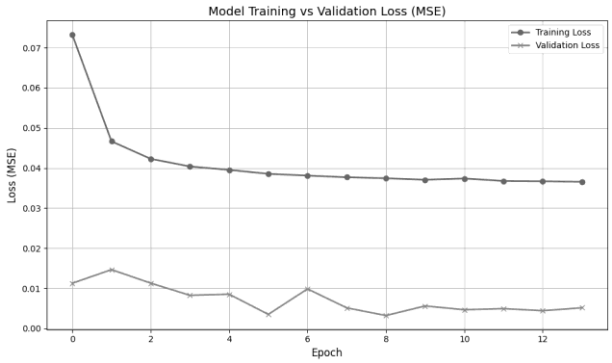

Fig 2. Training and validation loss of the ANN model

From figure 2, we can see that after epochs 9 the validation loss is starting to make a linear curve, meaning it has overcome the overfitting problem and model is generalizing the data and training it. The model is not learning the data but generalizing it.

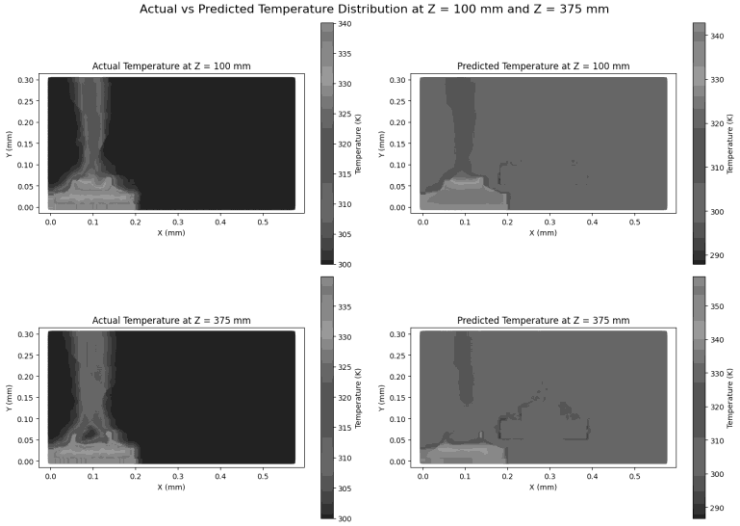

Fig 3. Plot of actual vs predicted temperature for z = 100 mm and 375 mm plane respectively

7. Challenges Faced

1) Huge number of 2d data: For 2d plane x and y coordinate with z = 0 plane there were 347755 data indexes.
2) Too few data for 3d plane: For 3d plane with different x and y coordinate along with z planes there were only 4899 data indexes.
3) X and Y coordinates not matching: For prediction of temperature model needed the same x and y coordinates in the 2d and 3d data sets.
4) Overfitting in ANN: Early models overfitted training data, especially when model depth was increased.
5) Mesh Resolution vs. Model Complexity: In ANSYS the 2d plane was made with fine mesh and 3d plane was made with coarse mesh generating irrelevancy in data indexes.

7.1 Solutions Implemented:

1. Data Augmentation:
   Various data augmentation techniques like scaling methods were applied to improve the robustness and generalization of the model.

2. Dropout and Batch Normalization:

   A 20% dropout was introduced in the ANN architecture to prevent overfitting.

   Batch normalization was added to stabilize and accelerate the training process.

3. 3D Data Up-Sampling:

   Interpolation of 3D temperature data was performed using the Inverse Distance Weighting (IDW) and cKDTree algorithms.

   The interpolation was done on the same (x, y) grid as the 2D data to maintain alignment along same indexes with 10 neighbours and 2^{nd} power of the Euclidean distance.

4. Computational Setup:

   GPU acceleration on the local machine was utilized to speed up training process.

   Jupyter Notebook with ipykernel was integrated into VS Code for efficient model development and experimentation.

```
df_2d = df_visualize_2d[['x', 'y']].dropna()
points_2d = df_2d[['x', 'y']].values

k_neighbors = 10
power = 2

target_z_planes = [100, 375]
output_folder = r"S:\Nit Durgapur\College 4th Sem\JU Internship\Fins Heat Prediction\ANN Model\3D Test Interpolated Data"

for z_plane in target_z_planes:
    df_3d_current = df_visualize_3d[df_visualize_3d['Z'] == z_plane][['x', 'y', 'Temperature']].dropna()

    points_3d_current = df_3d_current[['x', 'y']].values
    values_3d_current = df_3d_current['Temperature'].values

    tree = cKDTree(points_3d_current)
    distances, indices = tree.query(points_2d, k = k_neighbors)

    weights = 1.0 / (distances**power + 1e-9)
    weights /= weights.sum(axis = 1)[ :, np.newaxis]

    temp_interpolated = np.sum(values_3d_current[indices] * weights, axis = 1)

    df_current_interpolated = pd.DataFrame({
        'X': df_2d['x'].values,
        'Y': df_2d['y'].values,
        'Z': z_plane,
        'Temperature_Interpolated': temp_interpolated
    })

    output_file = os.path.join(output_folder, f'z{z_plane}_upsample_interpolated_data.csv')
    df_current_interpolated.to_csv(output_file, index = False)
    print(f"✅ Saved: z{z_plane}_upsample_interpolated_data.csv")

print("\n🎯 Interpolation for Z = 100 mm and Z = 375 mm completed!")
```

Fig 4. The code snippet of IDW interpolation is depicted here

7.2 Equations Used:

Equation 6 shows the Euclidean distance which is given by d_{ji} between j^{th} and i^{th} position

$$d_{ji} = \sqrt{\left(x_j - x_i\right)^2 + \left(y_j - y_i\right)^2} \qquad (6)$$

Equation 7 is weights for its k-nearest neighbor.

$$W\_ji = \frac{1}{\left(d_{ji}^{P} + \epsilon\right)} \qquad (7)$$

Normalization of the weights is given by Eqn. 8.

$$\widetilde{w_j}i = \frac{w_j i}{\sum_{i=1}^{k} w_j i} \qquad (8)$$

Interpolation T_i at point j is at Eqn. 9

$$\widehat{T_j} = \sum_{i=1}^{k} w_{ji} \cdot T_i \qquad (9)$$

8. Key Learnings

The internship significantly enhanced my academic and practical skills:

- Gave insights how we work in an academic research lab.
- Developed a mindset to integrate conventional systems with AI approach.
- Gained confidence in using simulation software with coding environments.
- Learned how to approach real-world engineering problems using a structured ML workflow.
- Improved soft skills like documentation, visualization, and team communication.
- Learned how to evaluate AI models not just by scores but by how well they respect physical and real-world architectures.
- Understood the limitations of simulation-based design and how AI can enhance the upcoming thermal research field.

This experience shaped my thinking to not only be a mechanical engineer but also a problem-solver capable of using tools from multiple disciplines.

9. Conclusion

This internship at Neptune Lab, Jadavpur University, was a good and adverse experience where I got the chance to apply what I've learned in both mechanical engineering and artificial intelligence. Working on the project of predicting 3D temperature fields from just 2D data using Artificial Neural Networks taught me how powerful machine learning can be when combined with core engineering concepts.

Instead of relying only on traditional 3D simulations, which are often time-consuming and resource-heavy, I saw how a smart, well-trained model can offer accurate results in just short amount of time. This not only improved my technical skills in programming and data analysis but also helped me understand how to think critically about the physical relevance of AI models.

Throughout the internship, I got to explore tools like ANSYS, Python, and Jupyter Notebooks, and more importantly, I learned how to bring all of them together in a complete workflow—from simulation to prediction. I also realized the value of being able to communicate and present my work clearly, both through visuals and documentation.

More than just a technical project, this internship gave me a taste of what real-world research is like. It showed me how mechanical engineers can play an active role in future technologies by learning to collaborate with data scientists and adopting AI tools. This experience has encouraged me to keep exploring the intersection of simulation, coding, and intelligent design in my future studies and career.

9.1 Future Developments:

1) Depicting a universal feature, such as epsilon or delta which we could multiply with 2d data to get the 3d data. For example, $\in^* = \in/\varepsilon$ (where $\in$ = Temperature difference between 3d and 2d fins along some particular x and y coordinate and ε = Difference in z planes and $\in^*$ is the term to be multiplied with 2d temperature data along with different x, y coordinates to get the required 3d plane temperature data).
2) Developing more robust ML model and physically testing them in real world like XGBoost Classifier or Support Vector Regression according to the given dataset.
3) Reducing the time and monetory cost of the approach by getting one simple model to get the more accurate result without heavy and expensive resources.

10. References

1. K Kim, H Lee, M Kang, G Lee, K Jung, C R Kharangate, M Asheghi, K E Goodson, H Lee, A machine learning approach for predicting heat transfer characteristics in micro-pin fin heat sinks, 2022, 194, 123087, https://doi.org/10.1016/j.ijheatmasstransfer.2022.123087

2. H Lee, G Lee, K Kim, D Kong, H Lee, Multimodal machine learning for predicting heat transfer characteristics in micro-pin fin heat sinks, 2024, 57, 104331, https://doi.org/10.1016/j.csite.2024.104331

3. S Karmakar, A Mohanty, 2D Numerical Analysis of Natural Convection in Vertical Fins on Horizontal Base, and Fluid Flow, https://doi.org/10.1007/978-981-13-1903-7_48