Power, Area and Thermal Prediction in 3D Network-on-Chip using Machine Learning

Abhijith C, Anand M K
Department of Computer Science and Engineering
National Institute of Technology Karnataka (NITK)
Surathkal, India
Email: {abhijithc.242cs003, anandmk.242cs008}@nitk.edu.in

I. Experimental Results

This section focuses on the experimental setup, dataset generation, dataset preprocessing, performance of different models, and comparison of their performance.

A. Experimental Setup

The dataset is generated using PAT-Noxim, a cycle-accurate simulator, and a shell script. The entire experiment is executed on a computer setup with the configurations: HP HP EliteDesk 800 G8 Tower PC, 16.0 GiB memory, 11th Gen Intel® Core™ i5-11500 @ 2.70GHz × 12 graphics, Mesa Intel® Graphics (RKL GT1), 1.0 TB disk capacity and Ubuntu 22.04.4 LTS.

B. Dataset Generation

The dataset is generated by simulating various configurations on PAT-Noxim. The configurations include mesh sizes ranging from 2 x 2 x 2 to 16 x 16 x 2, pir values from 0.01 to 0.1 with step size of 0.01, and buffer sizes 4, 6, 8, and 10. Three different routing algorithms are used: XYZ, Fully Adaptive, and Odd-Even 3D. The simulations are run for 200000 cycles. The traffic considered is Random.

C. Data Preprocessing

The simulation results of PAT-Noxim include various metrics. The parameters considered in this experiment are power metrics such as average power, average core power, average router power, and average power per router; area metrics such as layer area, area per core, and total area; temperature metrics such as steady state temperature, core average temperature, memory average temperature, and router average temperature. The categorical column, such as the routing algorithm, is encoded. The dataset is split into training and test sets (80% train and 20% test). The parameters are standardized to the same scale, which improves the performance of ML models.

D. Models Used

The generated dataset is trained using the following models:

- Random Forest
- Decision Tree
- AdaBoost
- AdaBoost with Decision Tree
- Support Vector Regressor (SVR)
- Linear Regression
- K-Nearest Neighbors

E. Performance Metrics

The performance metrics used for evaluating the performance of different models are:

- **Mean Squared Error (MSE)**: Average of the squares of the difference between the predicted values and actual values..
- **Mean Absolute Error (MAE)**: Average of the absolute difference between the predicted values and actual values..
- **R^2 (Coefficient of Determination)**: The dependent variable variance proportion explained by the independent variables.

F. Results Analysis

1) Temperature Analysis: This section analyzes temperature-related parameters such as steady-state temperature, core average temperature, memory average temperature, and router average temperature. Tables I and II analyze layer one and layer two steady-state temperatures. Layer one and two core average temperature metrics are provided in Tables III and IV. Tables V and VI analyze layer one and layer two memory average temperatures. Layer one and two router average temperature metrics are provided in Tables VII and VIII.

TABLE I: Performance Metrics for Different Algorithms - steady_state_temp_L0

Algorithm	**MSE**	**MAE**	R^2
AdaBoost with Decision Tree	0.0011	0.0125	0.9989
Random Forest	0.0012	0.0154	0.9987
Decision Tree	0.0021	0.0171	0.9978
KNN	0.0380	0.1171	0.9604
SVR	0.0613	0.1250	0.9362
AdaBoost	0.1979	0.3712	0.7939
Linear Regression	0.4100	0.4848	0.5731

2) Power Analysis: This section analyzes power-related parameters such as average power, average core power, average power per router, and average router power. Table IX analyzes the average core power. Average power metrics are provided

TABLE II: Performance Metrics for Different Algorithms - steady_state_temp_L1

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0011	0.0127	0.9988
Random Forest	0.0012	0.0153	0.9987
Decision Tree	0.0020	0.0169	0.9979
KNN	0.0383	0.1175	0.9601
SVR	0.0622	0.1252	0.9353
AdaBoost	0.1931	0.3624	0.7990
Linear Regression	0.4123	0.4862	0.5709

TABLE III: Performance Metrics for Different Algorithms - core_avg_temp_L0

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0077	0.0415	0.9922
Random Forest	0.0069	0.0436	0.9930
Decision Tree	0.0106	0.0506	0.9892
AdaBoost	0.3740	0.4481	0.6203
KNN	0.2283	0.2017	0.7683
SVR	0.4741	0.2536	0.5188
Linear Regression	0.6070	0.4970	0.3839

TABLE IV: Performance Metrics for Different Algorithms - core_avg_temp_L1

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0034	0.0267	0.9966
Random Forest	0.0027	0.0265	0.9973
Decision Tree	0.0043	0.0307	0.9958
SVR	0.3460	0.2720	0.6582
KNN	0.2191	0.2000	0.7835
AdaBoost	0.2557	0.4098	0.7474
Linear Regression	0.9037	0.5865	0.1072

TABLE V: Performance Metrics for Different Algorithms - mem_avg_temp_L0

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0064	0.0432	0.9936
Random Forest	0.0051	0.0398	0.9949
Decision Tree	0.0082	0.0471	0.9918
KNN	0.4311	0.2576	0.5710
AdaBoost	0.1985	0.3481	0.8025
SVR	0.9742	0.3193	0.0304
Linear Regression	0.8829	0.4742	0.1213

TABLE VI: Performance Metrics for Different Algorithms - mem_avg_temp_L1

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0020	0.0212	0.9980
Random Forest	0.0017	0.0198	0.9983
Decision Tree	0.0030	0.0235	0.9971
KNN	0.2512	0.2029	0.7556
AdaBoost	0.1464	0.2762	0.8576
SVR	0.3990	0.2770	0.6118
Linear Regression	1.0030	0.5763	0.0242

TABLE VII: Performance Metrics for Different Algorithms - router_avg_temp_L0

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0047	0.0313	0.9952
Random Forest	0.0050	0.0362	0.9950
Decision Tree	0.0075	0.0413	0.9925
KNN	0.2341	0.1797	0.7651
AdaBoost	0.2079	0.3494	0.7915
SVR	0.5236	0.2204	0.4748
Linear Regression	0.5870	0.4415	0.4112

TABLE VIII: Performance Metrics for Different Algorithms - router_avg_temp_L1

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0020	0.0193	0.9981
Random Forest	0.0020	0.0211	0.9981
Decision Tree	0.0033	0.0244	0.9968
KNN	0.2300	0.1786	0.7807
AdaBoost	0.1616	0.3120	0.8459
SVR	0.3708	0.2496	0.6464
Linear Regression	0.9080	0.5579	0.1340

in Table X. Table XI analyzes the average power per router. Average router power metrics are provided in Table XII.

TABLE IX: Performance Metrics for Different Algorithms - avg_cores_power

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0004	0.0063	0.9996
Random Forest	0.0004	0.0076	0.9996
Decision Tree	0.0007	0.0081	0.9993
SVR	0.0027	0.0429	0.9974
KNN	0.0062	0.0614	0.9939
AdaBoost	0.0695	0.2264	0.9309
Linear Regression	0.1159	0.2551	0.8849

TABLE X: Performance Metrics for Different Algorithms - avg_power

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0007	0.0086	0.9993
Random Forest	0.0007	0.0101	0.9993
Decision Tree	0.0012	0.0108	0.9988
SVR	0.0029	0.0427	0.9971
KNN	0.0066	0.0611	0.9933
AdaBoost	0.1072	0.2854	0.8927
Linear Regression	0.1355	0.2651	0.8644

3) Area Analysis: This section analyzes area-related parameters such as layer area, total area, and area per core. Table XIII analyzes the layer area. Total area metrics are provided in Table XIV. Table XV analyzes the area per core.

G. Comparison study

1) Temperature Analysis: This section compares the performance of different models across parameters such as steady-state temperature, core average temperature, memory average temperature, and router average temperature. After studying

TABLE XI: Performance Metrics for Different Algorithms - avg_power_per_router

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0040	0.0204	0.9959
Random Forest	0.0079	0.0371	0.9919
KNN	0.0080	0.0368	0.9918
Decision Tree	0.0121	0.0393	0.9877
SVR	0.0143	0.0796	0.9854
AdaBoost	0.1188	0.2933	0.8785
Linear Regression	0.1873	0.3189	0.8085

TABLE XII: Performance Metrics for Different Algorithms - avg_routers_power

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0.0012	0.0123	0.9987
Random Forest	0.0023	0.0170	0.9976
Decision Tree	0.0034	0.0186	0.9965
SVR	0.0058	0.0494	0.9941
KNN	0.0085	0.0552	0.9913
AdaBoost	0.2717	0.4722	0.7219
Linear Regression	0.2406	0.3368	0.7537

TABLE XIII: Performance Metrics for Different Algorithms - layer_area

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	3.37E-32	2.77E-17	1.0000
Random Forest	8.04E-05	0.0029	0.9999
Decision Tree	0.0003	0.0025	0.9997
SVR	0.0041	0.0568	0.9960
KNN	0.0057	0.0594	0.9944
AdaBoost	0.0335	0.1462	0.9673
Linear Regression	0.1023	0.2403	0.9000

TABLE XIV: Performance Metrics for Different Algorithms - total_area

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	1.81E-10	6.88E-07	1.0000
Random Forest	8.04E-05	0.0029	0.9999
Decision Tree	0.0003	0.0025	0.9997
SVR	0.0041	0.0568	0.9960
KNN	0.0057	0.0594	0.9944
AdaBoost	0.0333	0.1469	0.9674
Linear Regression	0.1023	0.2403	0.9000

TABLE XV: Performance Metrics for Different Algorithms - area_per_core

Algorithm	MSE	MAE	R^2
AdaBoost with Decision Tree	0	0	1
AdaBoost	0	0	1
Random Forest	0	0	1
Decision Tree	0	0	1
SVR	0	0	1
KNN	0	0	1
Linear Regression	0	0	1

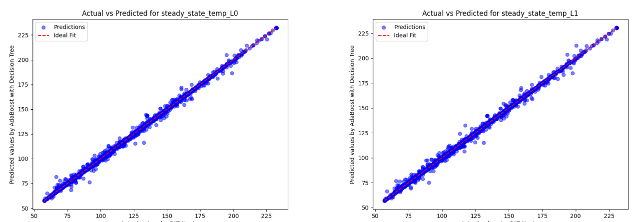
Tables I and II, it is evident that AdaBoost with Decision Tree and Random Forest achieved minimal errors (MSE, MAE) and the highest R^2 values ($\geq$0.998 $\geq$0.998) for both layers. Algorithms like KNN and SVR had higher errors and lower R^2 values. Tables III, IV, V, VI, VII, and VIII show that AdaBoost with Decision Tree and Random Forest performs well with minimal MSE and MAE values and R^2 values close to 1 across both layers. Linear regression and SVR show the worst performance.

2) Power Analysis: This section compares the performance of different models across parameters such as average power, average core power, average power per router, and average router power. After studying Table IX, it is evident that AdaBoost with Decision Tree and Random Forest show close to zero errors (MSE, MAE) and the highest R^2 values ($\geq$0.99). Algorithms like Linear regression and KNN had higher errors and lower R^2 values. Tables X, XII, and XI show that AdaBoost with Decision Tree performs better than all other algorithms with minimal MSE and MAE values and R^2 values close to 1 for average power metrics. Linear regression and KNN show the worst performance with relatively high errors.

3) Area Analysis: This section compares the performance of different models across parameters such as layer area, total area, and area per core. After studying Tables XIII and XIV, it is evident that AdaBoost with Decision Tree and Random Forest show near-perfect performance with close to zero errors (MSE, MAE) and R^2 values close to 1. Algorithms like SVR and KNN had higher errors and lower R^2 values. Table XV shows that all algorithms made perfect predictions with MSE and MAE values as 0 and R^2 value as 1 because the area per core is a constant value.

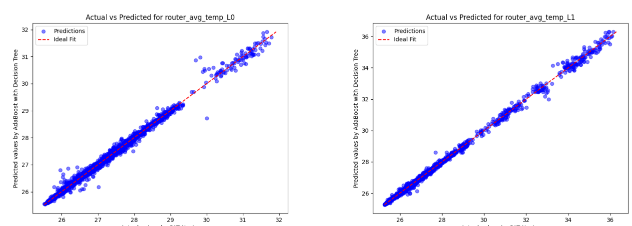
AdaBoost with Decision Tree and Random Forest consistently outperformed all other models with the lowest MSE and MAE values and the highest R^2 values across the prediction of all the power, area, and thermal metrics. Random Forest shows slightly less performance than AdaBoost with Decision Tree. Linear Regression and SVR show the worst performance with higher errors and low R^2 values. It is concluded that AdaBoost with Decision Tree is the most consistent and accurate model among all other models. This model works by training a series of 50 decision trees, each aimed at reducing the overall prediction error. Each round assigns higher weights to the samples, which are hard to predict. The subsequent trees give more importance to those challenging cases. The final prediction is the aggregation of each tree, which overall makes a robust prediction model.

Figures 1 to 7 compare actual values from PAT Noxim and predicted values from the AdaBoost model with the decision tree. The blue dots are individual data points from the test dataset, where each dot shows how closely the model's predictions match the actual values. The red dashed line represents the ideal fit where the model's predictions match the actual values. If points are closely clustered around the red line, this suggests that the model's predictions are relatively accurate.A larger spread from the red line indicates higher prediction errors. Figures 1 to 4 show that the actual and predicted values are almost identical for the temperature parameters. Temperature is measured in degrees Celsius. The blue points and red lines are nearly perfectly aligned. Figures 5 and 6 illustrate that the predicted and actual values are perfectly aligned, highlighting the effectiveness of the AdaBoost model with the decision tree in predicting power values. Power is calculated in (J/cycle).Figure 7 demonstrates that the area is perfectly predicted by the AdaBoost model with the decision tree. The area is measured in (μm^2).



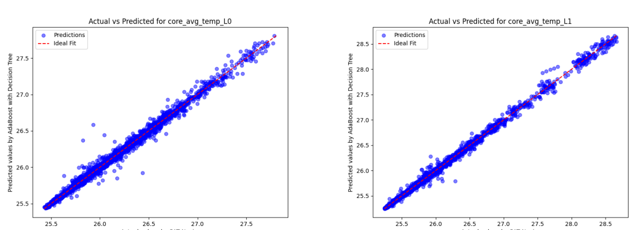
(a) Actual vs Predicted for steady_state_temp_L0 (b) Actual vs Predicted for steady_state_temp_L1

Fig. 1: Actual vs Predicted for steady_state_temp_L0 and steady_state_temp_L1



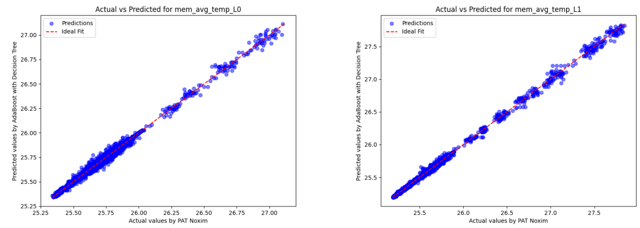
(a) Actual vs Predicted for router_avg_temp_L0 (b) Actual vs Predicted for router_avg_temp_L1

Fig. 2: Actual vs Predicted for router_avg_temp_L0 and router_avg_temp_L1



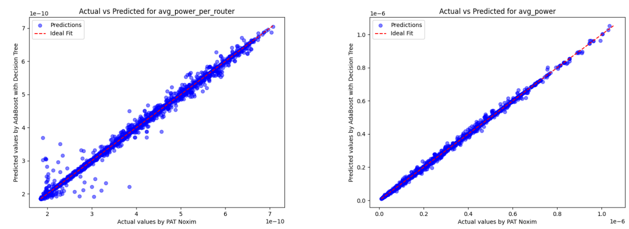
(a) Actual vs Predicted for core_avg_temp_L0 (b) Actual vs Predicted for core_avg_temp_L1

Fig. 3: Actual vs Predicted for core_avg_temp_L0 and core_avg_temp_L1



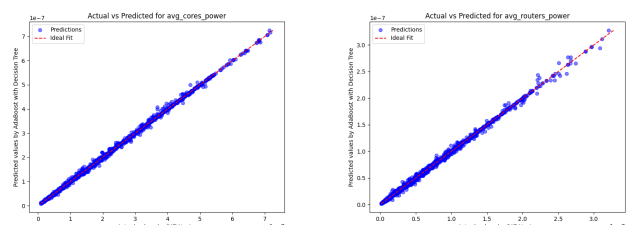
(a) Actual vs Predicted for mem_avg_temp_L0 (b) Actual vs Predicted for mem_avg_temp_L1

Fig. 4: Actual vs Predicted for mem_avg_temp_L0 and mem_avg_temp_L1



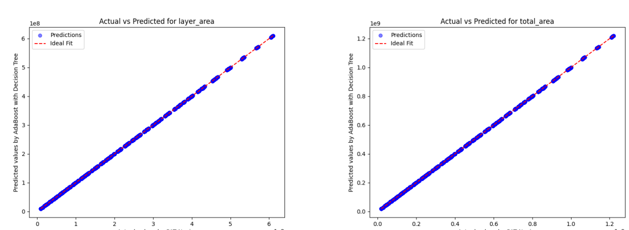
(a) Actual vs Predicted for avg_power_per_router (b) Actual vs Predicted for avg_power

Fig. 5: Actual vs Predicted for avg_power_per_router and avg_power



(a) Actual vs Predicted for avg_cores_power (b) Actual vs Predicted for avg_routers_power

Fig. 6: Actual vs Predicted for avg_cores_power and avg_routers_power



(a) Actual vs Predicted for layer_area (b) Actual vs Predicted for total_area

Fig. 7: Actual vs Predicted for layer_area and total_area