Artificial Intelligence Solution on Intel Xeon Processor Power and Performance Engineering

Zhongbin Liu
dept. Data Center Platform Engineering
Intel Asia Pacific
Shanghai, China
zhongbin.liu@intel.com

Xiaofei Jiang
dept. Data Center Platform Engineering
Intel Asia Pacific
Shanghai, China
xiaofei.jiang@intel.com

Jiajia Zhang
dept. Data Center Platform Engineering
Intel Asia Pacific
Shanghai, China
jiajia.zhang@intel.com

***Abstract*— Nowadays the major Cloud Service Providers (CSP) are critically setting up high performance infrastructures to meet cloud customers' various computing demands. To help CSP customers invest the right places building high-performance Xeon-based systems based on their specifical usages, Intel invests significant engineering resources on Xeon products power performance features design, development, and validation, while the engineering cost is huge and not scalable. This paper introduces an Artificial Intelligence (AI) solution named Bench Counselor in post-silicon power performance development and validation, it could suggest the most valuable hardware investment areas as per customer usage or benchmarking methodology, meanwhile reduce the engineering resources. Training AI model with historical Xeon processor performance results and system hardware configurations, the AI solution could efficiently assist power and performance engineers for outliers categorizing and debugging, also provide heuristics on the most valuable investment areas to get significant performance gain.**

Keywords—Intel Xeon Server Processor, Power and Performance engineering, validation, development, hardware configuration, Artificial Intelligence, Machine Learning, XGboost

I. Introduction

Intel Xeon server product Power and Performance (PnP) is crucial to Data Center market. The Platform PnP engineering team is responsible for developing and validating Xeon power and performance features such as SGX, TDX, Accelerators, etc. With the evolvement of Intel server products and the emerging demands from customers, PnP engineering team has enlarged PnP benchmarking results by two times per Xeon design request and customers' feedback in 2023 and will predictably continue to enlarge in the near future to cover more AI Accelerators. Based on the statistics, the platform PnP team generates thousands of benchmarking results for validation and development purposes every day, which require PnP engineers to review, and grade based on a variety of platform hardware and software configurations. This tremendous amount of data is extremely challenging for PnP engineers to validate.

Meanwhile, AI technologies are blooming in recent years and integrating into various industries and applications, revolutionizing the way to approach complex problems and tasks. In Machine Learning (ML) domain, a diverse array of models has been developed to address different regression tasks, such as Linear Regression Model[1], Random Forest[2], Gradient Boosting Machines (GBMs) , XGBoost[3], Support Vector Regression (SVR)[4]. And Deep Learning (DL) has been demonstrating remarkable success for regression analysis in recent years, such as Convolutional Neural Networks (CNN) [5].

Bench Counselor is an AI-based solution that utilizes XGBoost and Linear Regression techniques. It takes platform hardware configurations as input features and outputs benchmark performance prediction and feature importance ranking list. Bench Counselor makes two principal contributions to the enhancement of power and performance engineering.

- Bench Counselor provides insights to PnP development engineers on the most valuable research areas by ranking feature importance based on training dataset distribution. This provides heuristics of performance improvement from hardware's perspective for different benchmarking results.
- Bench Counselor assists PnP engineers on first level issue triage by categorizing benchmarking results into different severity levels by Absolute Percentage Error (APE), which is delta indicator of Bench Counselor performance prediction results and measured results. As shown in Fig. 1, Bench Counselor was integrated into the typical PnP data collection and analysis

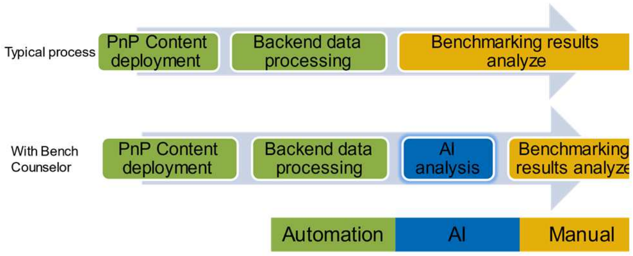

Fig. 1 PnP working flow with AI solution

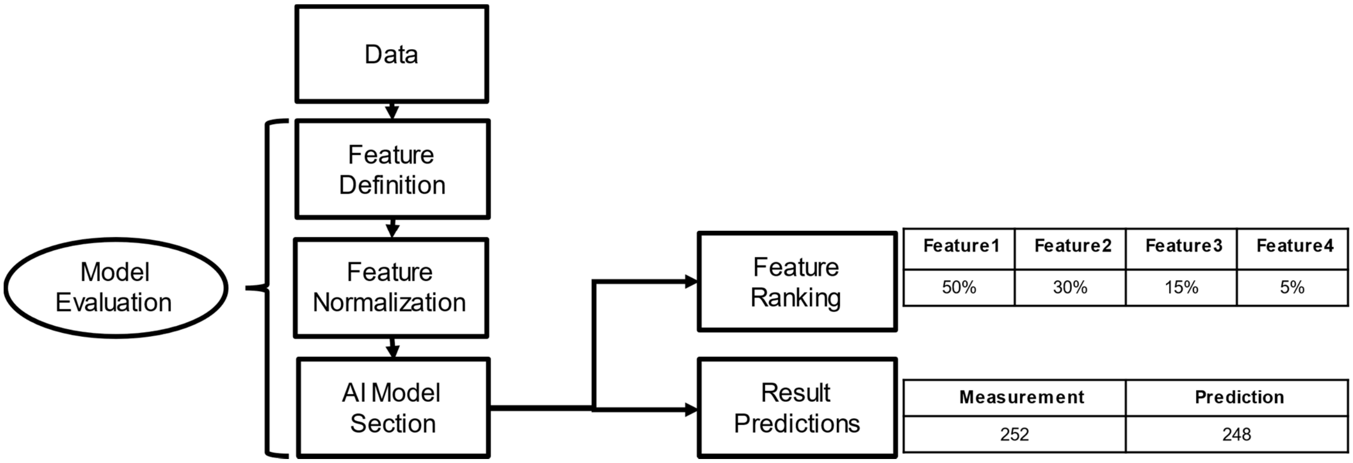

Fig. 2 Bench Counselor Structure Overview

process, with the purpose of improving debug efficiency by 10%.

Bench Counselor was firstly implemented on Intel 5th Xeon Scalable Processor codenamed Emerald Rapids (EMR). The model was trained with more than one hundred thousand of EMR benchmarking results associated with system hardware configurations. The configuration information was collected by Intel's internally developed debug tools, or from CPU specification documents. Verifying Bench Counselor on ten thousand of EMR Post Launch Release1 (PLR1) benchmarking results of 47 sub-workloads from 5 top industry benchmarks, 73.5% of EMR PLR1 benchmarking results were accurately classified as non-issue data, considering their Absolute Percentage Error (APE) between measurements and Bench Counselor predictions are less than 3%. This kind of non-issue data requires no or minimal PnP engineers' manual reviews, and hence PnP debug efficiency can be significantly improved. On the other hand, Bench Counselor can help PnP engineers to quickly locate results with big APEs for further debug. Under the assistance of Bench Counselor, several critical bugs were captured early on PLR1 before releasing to customers. Bench Counselor can also provide feature ranking for each sub-workload and the ranking generally aligns with PnP expert knowledge.

II. Description

The regular PnP data review and validation can be abstracted to a general Regression Task in the Machine Learning and Statistics domain, where multiple independent variables impact a dependent variable. In our case, system hardware configurations are independent variables and benchmarking result is the dependent variable.

The Bench Counselor structure, as shown in Fig. 2, consists of 5 main components: Data Annotation, Feature Definition, Feature Normalization, AI Model Selection and Model Evaluation. Bench Counselor has two outputs. One is prediction result per input features and the other is feature importance ranking result.

Besides Bench Counselor structure overview, the subsequent sections introduce Bench Counselor integration in Platform PnP Engineering.

A. Data Annotation

Training dataset is generated from 47 sub-workloads of 5 top industry-standard benchmarks on 300 EMR at-scale systems. The benchmarks include STREAM and Memory Latency Checker (MLC) for memory bandwidth and latency assessments, LINPACK and HPCG for linear algebra and matrix operations, SPECrate 2017 Integer and SPECrate 2017 Floating Point for comprehensive computing performance evaluation on integer and float point operations, with a variety of sub-workloads on different domains. Both the benchmarking results and hardware configurations are standardized and structured in database for training.

The data annotation module annotates the benchmark results with expected and unexpected labels. Dataset annotation is vital since it defines the roofline of the model's accuracy. To ensure the correctness of the dataset, PnP engineers are engaged to review and verify the entire dataset integrity to make sure measured benchmark results align with hardware configurations. It is infeasible to manually review all the data points to complete annotation, therefore a two-step approach is used: Coarse-Grained-Filter and Fine-Grained-Filter.

1) Coarse-Grained-Filter: The benchmarking results using as training data from at-scale systems has already been reviewed by debug engineers during previous validation and the unexpected results have been indentified. This kind of debug information can be leveraged to create a bunch of rules in the database to filter out the data with performance issues. Hence a clean view of data can be created in the database as the output of the Coarse-Grained-Filter.

2) Fine-Grained-Filter: Upon the clean view of the data from the Coarse-Grained-Filter, 5-fold cross-validation is deployed. The benchmarking results with APE greater than 3% are filtered out and sent to debug engineers for a second review.

Table 1 Feature List

CPU.Microarchitecture	CPU.TMUL All Core Turbo Freq Rate
CPU.TDP	CPU.TMUL Deterministic P1 Freq Rate
CPU.Core(s) per Socket	CPU.CLM P1 Max Freq
CPU.Socket(s)	CPU.CLM P0 Max Freq
CPU.Maximum Frequency	CPU.Max UPI Port Cnt
CPU.All-core Maximum Frequency	DIMM.XDPC
CPU.Base Frequency	DIMM.Total
CPU.L1d Cache Per Core	DIMM.Quantity
CPU.L1i Cache Per Core	DIMM.FirstPartNo
CPU.L3 Cache Per Socket	DIMM.OpsFreq
CPU.AVX2 All Core Turbo Freq Rate	DIMM.FirstDIMMSize
CPU.AVX3 Deterministic P1 Freq Rate	DIMM.RankType
CPU.AVX3 All Core Turbo Freq Rate	BIOS.NUMA Node(s)
WorkloadPreset	

By combining Coarse-Grained-Filter and Fine-Grained-Filter, only around fifteen thousand data points out of over one hundred thousand benchmarking results still need human efforts to review.

B. *Feature Definition*

The system hardware configurations are collected by Intel internally developed debug tools or from CPU specification documents. Feature selection module was designed to select these hardware configurations as AI model input features.

Including all hardware configurations as input features could overburden AI models and thus decrease inference accuracy as many of features may not be relevant to benchmarking results or duplicated with equivalent features. Therefore identifying, and combining equivalent features are essential to secure AI model prediction quality. We apply 2 steps to define the features.

1) Step1: PnP Engineers provide an initial list of hardware configurations that could be relevant to performance results based on their expertise. This helps shorten the input feature number to approximately 40.

2) Step2: These input features are continually optimized by removing duplicated features or splitting single feature into multiple features to make it precise and easy to adopt by AI models.

As shown in Table *1*, there are 27 hardware configurations finalized as input features, including CPU specifications and DIMM specifications on the systems.

C. *Feature Normalization*

The 27 input features are of different data types, including formats, range, distribution and units. To improve the model convergency, we decide to normalize features to float format and the following 5 methods are considered as candidates.

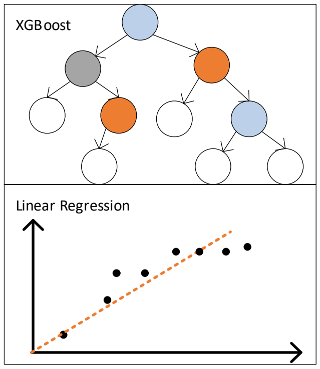

Fig. 3 ML models under evaluation

1) StandardScaler

2) Normalizer

3) MinMaxScaler: e.g. [-1,0,1]-> [0,0.5,1]

4) MinMaxLabelEncoder: e.g. ["A", "B", "C"]-> [0,0.5,1]

5) OneHotEncoder: change each unique element into a category. E.g. ["A", "B", "C"]-> [[0 0 1], [0 1 0], [1 0 0]]

Based on a series of our experiment results, MinMaxScaler method is chosen for integer and float features. For string features, we slightly modified the MinMaxLabelEncoder method to ensure the usability of the model in the production environment.

In the production environment, it's imperative to have the ability to do the inference properly with any data that may or may not be seen in the training dataset. For example, the inference may get data with a new 'DIMM.FirstPartNo' that was never seen in the training and we still don't want the model to crash. So Bench Counselor expands the feature category set of string type features by incorporating a special placeholder category, mapped to -1 after MinMaxLabelEncoder. This strategy allows Bench Counselor to normalize any novel features encountered during inference and thereby maintaining consistent inference of data beyond the initial training scope. Besides, using this strategy we can gradually add these new data points back to training dataset to continuously upgrade the model in a seamless way and provide a transparent experience for users without annoying failures.

Finally, the input to use after normalization is a matrix full of float values, with size of [dataset_size, feature_list] and values within [-1,1].

D. *Model Selection*

There are 2 AI models under evaluation as shown in Fig. 3: XGBoost and Linear Regression.

XGBoost (short for Extreme Gradient Boosting) stands out as a powerful and widely used ML model in regression and feature ranking tasks. We use the following configurations to setup an XGBoost model:

1) max_depth=10

2) n_estimators=80

3) learning_rate=0.1

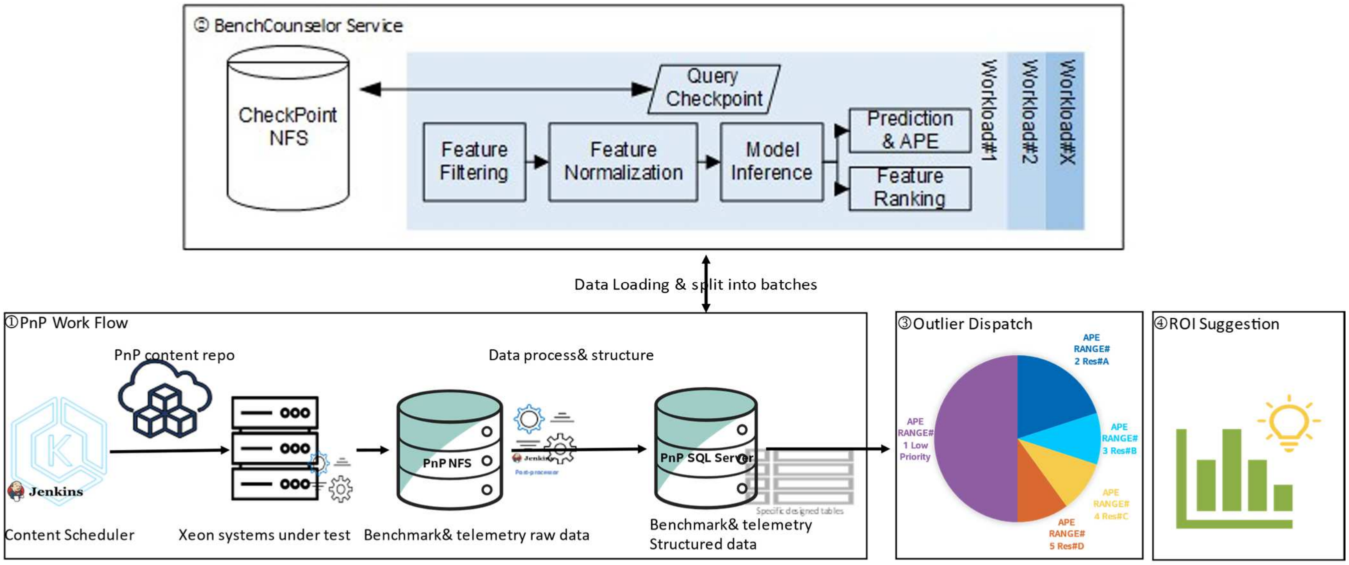

Fig. 4 Bench Counselor Integration in Platform PnP Engineering

4) booster="gbtree"

5) njobs=10

6) random_state=1

The feature ranking is generated by the "gain" metric, it measures the contribution of each feature to the improvement of the XGBoost model's performance during the feature split decisions.

Linear Regression model is one of the simplest and broadly used techniques. It offers an easy to implement and interpretable model. However, the linear regression's simplicity could be a limitation, particularly when dealing with complex, non-linear relationships in the data. Linear Regression model is taken as a baseline model and its formula is as (1), where each feature contribution is represented by weight w_x.

$$\hat{y}\left(w,\, x\right) \;=\; w_0 \;+\; w_1\, x_1 \;+\; \dots \;+\; w_p\, x_p \qquad (1)$$

E. Model Evaluation

As a regression task, metrics are required to evaluate the regression accuracy and keep optimizing Bench Counselor parameters. APE is defined for each benchmarking result for prediction error evaluation as shown in (2). Metric accuracy[α] is defined for the overall sub-workload accuracy evaluation during validation as shown in (3), where N refers to the benchmark results counts and α is 3% in Bench Counselor.

$$APE = abs\left(\frac{prediction - measurement}{measurement}\right) \qquad (2)$$

$$accuracy[\alpha] = \frac{N_{APE \leq \alpha}}{N_{total}} \qquad (3)$$

F. Bench Counselor Integration in Platform PnP Engineering

Bench Counselor inference module as an AI data processing engine has been integrated into the platform PnP working flow to enable regular PnP outlier triage as shown in Fig. 4.

1) PnP Work Flow: PnP benchmarks are scheduled to System Unter Test (SUT) for execution. SUTs collect system configurations and execute the benchmarks. After execution completed, the system configuration and benchmark logs are uploaded to NFS for post-processing. The bencmark results and system configuration data are finally post-processed to structured data format and uploaded to PnP SQL database.

2) Bench Counselor Service: Bench Counselor interacts with the SQL database bi-hourly to retrieve the recent data, including benchmarking results and system hardware configurations. The training and inference are completed per each of the sub-workloads to secure the feature ranking and benchmarking results prediction accuracies. For benchmarking results inference, it takes approximate two minutes on average to predict one hundred benchmarking results. Both benchmark prediction and APE values are written back to PnP SQL database.

3) Outlier Dispatch: APE values are used for benchmark result categorization based on the value range. For instance the benchmarking results with APE less than 3% are categorized to Range#1, which means results are meeting expectation and should be set as low priority. The benchamrking results with larger APE values are taken as outliers and will be assigned to different debug engineers based on the APE ranges.

4) Return of Investment (ROI) Suggestion: Per sub-workload feature ranking results are also recorded in database. They reflect each benchmarks' sensitivity to input features with weight numbers. The top ranking features are Return Of Investment(ROI) recommendations for each benchmark results.

III. Results & Discussion

Bench Counselor was initially trained and evaluated on EMR pre-PLR1 data, collected on massive systems of

Table 2 XGBoost and Linear comparison on validation dataset with Accuracy[3%] and APE>100% metrics

Benchmark Name	**Accuracy[3%]**		**APE>100%**	
	XGBoost	***Linear***	***XGBoost***	***Linear***
Stream	99.41%	62.36%	0.00%	0.00%
SPECrate2017_int_base	82.29%	77.81%	0.00%	0.10%
SPECrate2017_fp_base	82.64%	71.31%	0.00%	0.00%
MLC	88.76%	58.17%	0.10%	0.10%
Linpack	78.78%	55.76%	0.20%	0.80%
HPCG	91.48%	53.91%	0.00%	0.30%

different system hardware configurations. Then on EMR PLR1, Bench Counselor was used to demonstrate how much it can help on debug efficiency improvement and how the feature ranking results map to PnP engineering expertise.

In this section, we firstly present the Prediction Accuracy of AI models. Then we focus on outlier dispatch to improve Debug Efficiency and finally explore feature ranking for ROI suggestion.

A. *AI Model Prediction Accuracy*

Five-fold cross-validation method was used in the Bench Counselor model training to fully evaluate training dataset accuracy and fine-tune hyper-parameters. The training dataset was split into 5 folds, each containing a set of feature combinations that are intentionally non-overlapping with those in any other fold. This strategic approach to fold division ensures that when we train on four folds and validate on the remaining one, the validation results are not artificially inflated by the model's prior exposure to the same feature combinations during training.

As shown in Table 2, comparing the percentage of all benchmark predictions land into accuracy[3%], XGBoost is significantly performing better than Linear model across all benchmarks. Meanwhile XGBoost shows less far mis-predicted outliers with APE greater than 100%.

The result shows XGboost can generate more accurate predictions and make less mistakes. Thus, XGBoost model was selected as the primary ML model for next experiment and deployment.

B. *Debug Efficiency Improvement and Cost Reduction*

Based on the statistics in Fig. 5, the use of Bench Counselor on EMR PLR1 at-scale PnP data analysis is very helpful. 73.5% of benchmarking results show APE values less than 3% and these data points were taken as non-issue data and deprioritized to no manual data analysis effort or minimum manual effort. 23.4% of results show APE values between 3% to 40% and these data points were marked as high-risk area and assigned to experienced PnP engineers. Any benchmarking results with APE values over 40% are not expected to be a PnP issue and these data points were firstly assigned to junior debug engineers or function debug teams for error clearance.

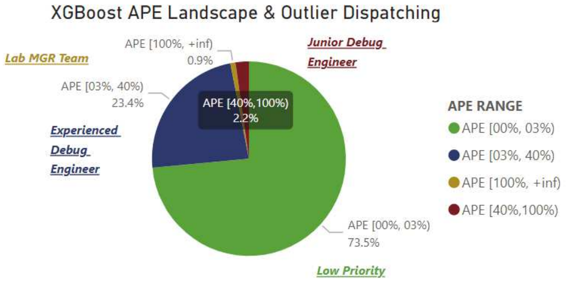

Fig. 5 XGBoost implementation APE landscape on EMR PLR1 benchmarking results

The results show that Bench Counselor has the key capability to enhance debug efficiency by flagging likely expected outcomes and potential setup errors automatically. At the same time, it can also assist PnP engineers to swiftly concentrate on critical issues by emphasizing key data indicative of potential bugs. Bench Counselor evaluation results on EMR PLR1 validation is positive to achieve the goals of efficiency improvement or labor cost reduction by 10%.

C. *Feature Ranking for Investment Suggestion*

The 47 sub-workloads of the benchmarks tested are categorized into 4 categories based on benchmarking methodologies and configuration sensitivities: Memory Bandwidth Sensitive, Latency Sensitive, Computing Sensitive and Mixed Sensitive. The top five ranking features and importance weight summary for each category is shown in Fig. 6, where columns represent the feature show up percentage as the top five feature in the sub-workloads of that category and the dots represent the average importance weight of the feature.

- For Memory Bandwidth Sensitive category, DIMM.Rank, CPU.CLM P0 Max Freq, DIMM.OpsFreq and DIMM.Quantity, which represents the rank for memory module, the max frequency of CPU Mesh Frequency, memory operation frequency, and the quantity of installed memories respectively, are ranking as top five vital features to benchmark results. This aligns with human expertise and expectation. However, the CPU.TDP is ranking as secondly important feature with 51.5% weight here, which is not expected. CPU.TDP feature represents the overall power budget of the processor and is not expected to be directly relevant to memory performance results in deep level. This could be due to limited AI model capability or feature correlation in training dataset.
- For Latency Sensitive category, CPU.CLM P0 Max Freq, DIMM.OpsFreq and DIMM.Quantity and DIMM. FirstPartNo are four of the top five features. Here DIMM. FirstPartNo represents the part number of the memory and contains the memory vendor and manufacturing information. However, DIMM.FirstDIMMSize, which represents the size of a single memory is not expected in the top five features. This could also be due to feature correlation.

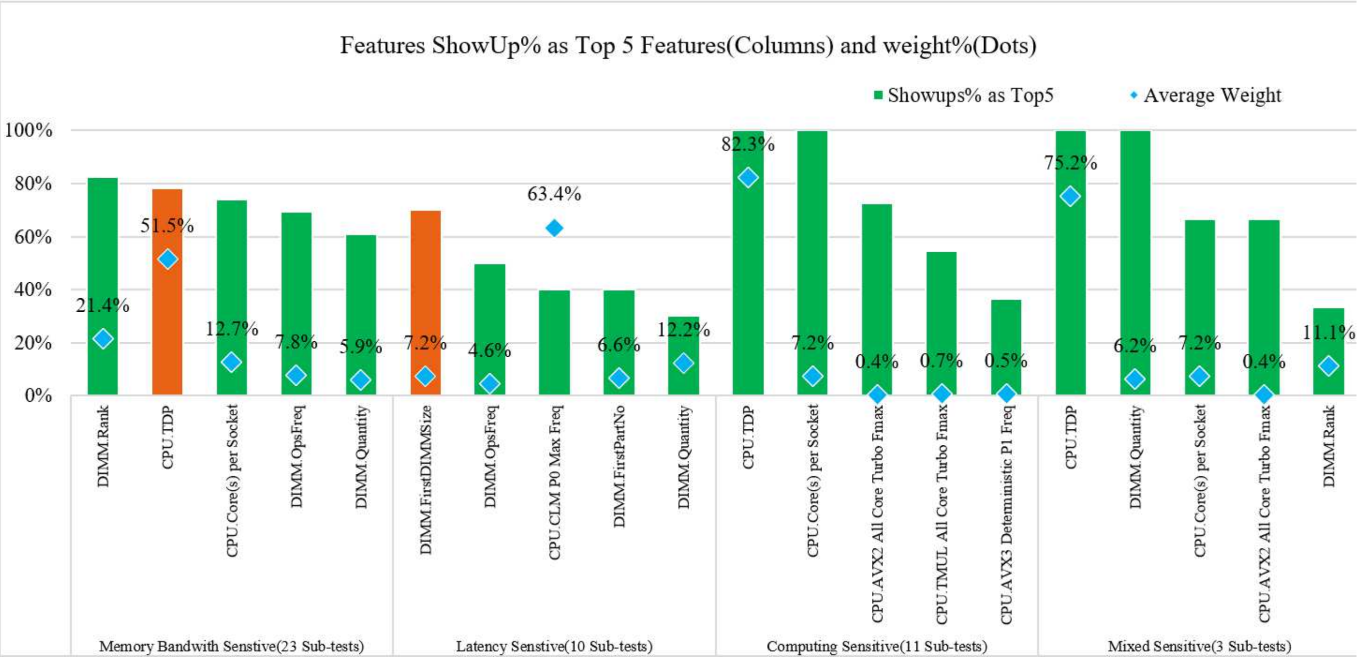

Fig. 6 Top5 Features ShowUp% and weights% Distribution among 4 categories.

- For Computing Sensitive category, CPU.TDP, CPU.Core(s) per Socket, CPU frequencies under different instruction set are of vital importance. The results generally align with human PnP expertise. A small misalignment is CPU.TDP weighs 82% which is too high and CPU.Core(s) is expected to weigh similar as CPU.TDP.
- For Mix Sensitive sub-workload category, CPU.TDP, CPU.Maximum.Freq, DIMM.Rank and DIMM.Quantity are top5 features, which also aligns with PnP expertise.

The top five feature ranking and weights are meaningful and mostly align with PnP expertise. They are good references for the hardware investment area in customer designs.

IV. Conclusion and Future Plans

A. Conclusion

Based on evaluation results on EMR PLR1 validation, Bench Counselor is proven capable to quickly and automatically rule out the good data from debug, locate and assign PnP outliers to proper engineering resources for debug efficiency improvement and cost reduction. 73.5% of all benchmarking results in EMR PLR1 were filter out as non-issues data which requires minimum human efforts, significantly improving debug efficiency and reducing the engineering cost. Bench Counselor also provides meaningful feature ranking for customer hardware investment suggestion.

B. Future Plan

The initial inference accuracy and feature ranking list are encouraging, while it is understood that there are still amounts of opportunities to further improve Bench Counselor design and engineering usage as the following:

- Enlarge integrated training dataset size. Create a test dataset for further AI model comparison and evaluation.
- Further research feature distribution and explore feature normalization methods such as Principal components analysis (PCA) to assist improving model training accuracy.
- Deep learning models are good at large-scale data, automated feature engineering, and non-linear relationship capturing. We are enabling CNN to leverage Deep Learning power for regression task.
- Add more hardware and software configurations, core Instruction Per Cycle (IPC), etc. to improve Bench Counselor capability.

References

[1] D. C. Montgomery, E. A. Peck, and G. G. Vining, "Introduction to Linear Regression Analysis," 5th ed. Hoboken, NJ: Wiley, 2012.

[2] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5-32, 2001.

[3] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2016, pp. 785-794.

[4] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in Adv. Neural Inf. Process. Syst., 1997, pp. 155-161.

[5] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proc. IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.