International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

COMPARATIVE ANALYSIS OF MACHINE LEARNING MODELS FOR PERFORMANCE PREDICTION OF THE SPEC BENCHMARKS

Madhu Bandari, . M. Sai Prasad Reddy

1. Assistant Professor, Department of Data Science and Artificial Intelligence, IcfaiTech School, Telangana, India. madhu.bandari@ifheindia.org

2. Department of Computer Science and Engineering, IcfaiTech School, Telangana, India. medipallysaiprasad@gmail.com

ABSTRACT: Simulation-based performance prediction is cumbersome and time-consuming. An alternative approach is to consider supervised learning as a means of predicting the performance scores of Standard Performance Evaluation Corporation (SPEC) benchmarks. SPEC CPU2017 contains a public dataset of results obtained by executing 43 standardised performance benchmarks organised into 4 suites on various system configurations. This paper analyses the dataset and aims to answer the following questions: I) can we accurately predict the SPEC results based on the configurations provided in the dataset, without having to actually run the benchmarks? II) what are the most important hardware and software features? III) what are the best predictive models and hyperparameters, in terms of prediction error and time? and IV) can we predict the performance of future systems using the past data? We present how to prepare data, select features, tune hyperparameters and evaluate regression models based on Multi-Task Elastic-Net, Decision Tree, Random Forest, and Multi-Layer Perceptron neural networks estimators. Feature selection is performed in three steps: removing zero variance features, removing highly correlated features, and Recursive Feature Elimination based on different feature importance metrics: elastic-net coefficients, tree-based importance measures and Permutation Importance. We select the best models using grid search on the hyperparameter space, and finally, compare and evaluate the performance of the models. We show that tree-based models with the original 29 features provide accurate predictions with an average error of less than 4%. The average error of faster Decision Tree and Random Forest models with 10 features is still below 6% and 5% respectively.

Keywords *– Machine learning , performance analysis , predictive models , SPEC CPU2017 , supervised learning*

1. INTRODUCTION

Using Machine Learning (ML) to improve system design and predict the performance of computer systems is an active research area [1]. System designers and engineers use prediction results to investigate the impact of configuration changes on system performance and make better design decisions. Vendors seek the best way to position their systems in the market before they are built. Thus, performance prediction of upcoming system configurations is a demanding task [2]. Moreover, consumers try to improve the cost-performance ratio by searching for the best configurations to optimise performance of their systems, or make rational

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

purchasing decisions. Having access to performance results of various workloads, even on large collections of computer systems is not enough, as consumers may require the performance data for new systems or unseen configurations [3]. These challenges motivate the study of performance prediction and evaluation. Nonetheless, accurate performance prediction of unseen configurations in terms of execution time, or throughput (when running concurrent jobs), is challenging. It may involve precise analytical modelling of future systems, which has become increasingly complicated with the advances in computer architecture. Also, modelling techniques relying on exhaustive and timeconsuming simulations might not even result in the most accurate predictions [2], [4]. Thus, rather than fine-grained system modelling [5], we rely on regression models for performance prediction. Such models learn the relationships between the configurations of the systems and their performance for various workloads. SPEC CPU2017 contains suites of industry-standard compute-intensive benchmarks, mainly considering the processor characteristics, memory subsystems, and compilers. SPEC provides real-world and portable programs which solve problems of various sizes [6] and are divided into four benchmark suites: 1) Floating Point rate: FP_rate, 2) Floating Point speed: FP_speed, 3) Integer rate: Int_rate, and 4) Integer speed: Int_speed. This paper uses the published performance results of the SPEC CPU2017 public dataset, and develops supervised learning models based on hardware and software features extracted from the computer systems benchmarked in that dataset. Our models are based on Multitask Elastic-Net (MT_EN), Decision Tree (DT), Random Forest (RF), and Multi-layer Perceptron (MLP) estimators.

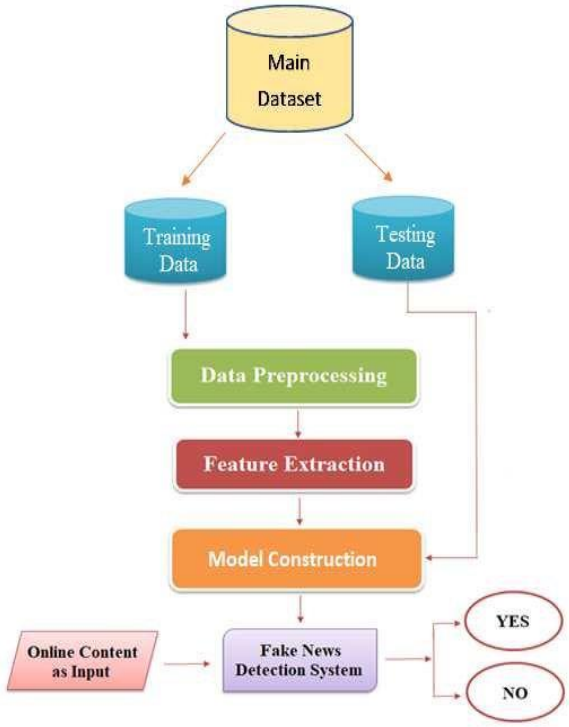

Fig.1: Example figure

Previous studies have focused on the accuracy of prediction of the SPEC benchmarks, mostly using neural networks. On the other hand, the importance of the features in terms of their contribution to the prediction models has been overlooked or neglected. The aim of our study is to build an ML pipeline in order to develop fast and accurate regression models for performance prediction of SPEC CPU2017, and provide a streamlined procedure for comprehensive evaluation. One of the main contributions of this work is to identify the importance of several hardware and software features, and compare the prediction error and latency of various models on the full and selected feature sets. This study also tries to uncover whether the data from existing systems (past data) can be used to predict the performance of future systems. The open source code is written in Python and relies on the scikit-learn [7] ML library. The ML pipeline and analysis

provided have some practical implications too, e.g. they could help engineers to reduce the design space and consumers to consider more important factors when making purchasing decisions.

2. LITERATURE REVIEW

A survey of machine learning for computer architecture and systems:

It has been a long time that computer architecture and systems are optimized for efficient execution of machine learning (ML) models. Now, it is time to reconsider the relationship between ML and systems, and let ML transform the way that computer architecture and systems are designed. This embraces a twofold meaning: improvement of designers' productivity, and completion of the virtuous cycle. In this paper, we present a comprehensive review of the work that applies ML for computer architecture and system design. First, we perform a high-level taxonomy by considering the typical role that ML techniques take in architecture/system design, i.e., either for fast predictive modeling or as the design methodology. Then, we summarize the common problems in computer architecture/system design that can be solved by ML techniques, and the typical ML techniques employed to resolve each of them. In addition to emphasis on computer architecture in a narrow sense, we adopt the concept that data centers can be recognized as warehouse-scale computers; sketchy discussions are provided in adjacent computer systems, such as code generation and compiler; we also give attention to how ML techniques can aid and transform design automation. We further provide a future vision of opportunities and potential directions, and envision that applying ML for computer architecture and systems would thrive in the community.

SPECNet: Predicting SPEC scores using deep learning:

In this work we show how to build a deep neural network (DNN) to predict SPEC® scores – called the SPECnet. More than ten years have passed since the introduction of the SPEC CPU2006 suite (retired in January 2018) and thousands of submissions are available for CPU2006 integer and floating point benchmarks. We build a DNN which inputs hardware and software features from these submissions and is subsequently trained on the corresponding reported SPEC scores. We then use the trained DNN to predict scores for upcoming machine configurations. We achieve 5%-7% training and dev/test errors pointing to pretty high accuracy rates (93%-95%) for prediction. Such a prediction rate is very comparable to expected human-level accuracy of 97%-98% achieved via careful performance modelling of the core and un-core system components. In addition to the CPU2006 suite, we also apply SPECnet to SPEComp2012 and SPECjbb2015. Though the reported submissions for these benchmark suites number in hundreds only, we show that such a DNN is able to predict for these benchmarks reasonably well (~85% accuracy) too. Our SPECnet implementation uses state-of-the-art Tensorflow infrastructure and is extremely flexible and extensible.

Predicting new workload or CPU performance by analyzing public datasets:

The marketplace for general-purpose microprocessors offers hundreds of functionally similar models, differing by traits like frequency, core count, cache

size, memory bandwidth, and power consumption. Their performance depends not only on microarchitecture, but also on the nature of the workloads being executed. Given a set of intended workloads, the consumer needs both performance and price information to make rational buying decisions. Many benchmark suites have been developed to measure processor performance, and their results for large collections of CPUs are often publicly available. However, repositories of benchmark results are not always helpful when consumers need performance data for new processors or new workloads. Moreover, the aggregate scores for benchmark suites designed to cover a broad spectrum of workload types can be misleading. To address these problems, we have developed a deep neural network (DNN) model, and we have used it to learn the relationship between the specifications of Intel CPUs and their performance on the SPEC CPU2006 and Geekbench 3 benchmark suites. We show that we can generate useful predictions for new processors and new workloads. We also cross-predict the two benchmark suites and compare their performance scores. Œe results quantify the self-similarity of these suites for the first time in the literature. Œis work should discourage consumers from basing purchasing decisions exclusively on Geekbench 3, and it should encourage academics to evaluate research using more diverse workloads than the SPEC CPU suites alone

Machine learning models to predict performance of computer system design alternatives:

Computer manufacturers spend a huge amount of time, resources, and money in designing new systems and newer configurations, and their ability to reduce costs, charge competitive prices, and gain market share depends on how good these systems perform. In this work, we concentrate on both the system design and the architectural design processes for parallel computers and develop methods to expedite them. Our methodology relies on extracting the performance levels of a small fraction of the machines in the design space and using this information to develop linear regression and neural network models to predict the performance of any machine in the whole design space. In terms of architectural design, we show that by using only 1% of the design space (i.e., cycle-accurate simulations), we can predict the performance of the whole design space within 3.4% error rate. In the system design area, we utilize the previously published Standard Performance Evaluation Corporation (SPEC) benchmark numbers to predict the performance of future systems. We concentrate on multiprocessor systems and show that our models can predict the performance of future systems within 2.2% error rate on average. We believe that these tools can accelerate the design space exploration significantly and aid in reducing the corresponding research/development cost and time-to-market.

A survey on multi-output regression:

In recent years, a plethora of approaches have been proposed to deal with the increasingly challenging task of multi-output regression. This paper provides a survey on state-of-the-art multi-output regression methods, that are categorized as problem transformation and algorithm adaptation methods. In addition, we present the mostly used performance evaluation measures, publicly available data sets for multi-output regression real-world problems, as well as open-source software frameworks.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

3. METHODOLOGY

Previous studies have focused on the accuracy of prediction of the SPEC benchmark, mostly using neural networks. Also, the importance of features in terms of their contribution to the models has been overlooked or neglected. The aim of our study is to build an ML pipeline to develop fast and accurate regression models for performance prediction of SPEC CPU2017, and provide a comprehensive evaluation of the results. One of the main contributions of this work is to identify the importance of several hardware and software features, and compare the prediction error and latency of various models on the full and reduced feature sets.

Drawbacks:

1. Accurate performance prediction of unseen configurations in terms of execution time or throughput (when running concurrent jobs) is challenging.
2. Modelling techniques relying on exhaustive and time-consuming simulations might not even result in the most accurate predictions.

In this study, we apply supervised learning to predict the performance scores of Standard Performance Evaluation Corporation (SPEC) benchmarks. The SPEC CPU2017 is a public dataset of results obtained by executing 43 standardised performance benchmarks organised into 4 suites on various system configurations. This paper analyses the dataset and aims to answer the following questions: I) can we accurately predict the SPEC results based on the configurations provided in the dataset, without having to actually run the benchmarks? II) what are the most important hardware and software features? III) what are the best predictive models and hyperparameters, in terms of prediction error and time? and IV) can we predict the performance of future systems using the past data? We present how to prepare data, select features, tune hyperparameters and evaluate regression models based on Multi-Task Elastic-Net, Decision Tree, Random Forest, and Multi-Layer Perceptron neural networks estimators.

Benefits:

1. We select the best models using grid search on the hyperparameter space, and finally, compare and evaluate the performance of the models.
2. We show that tree-based models with the original 29 features provide accurate predictions.

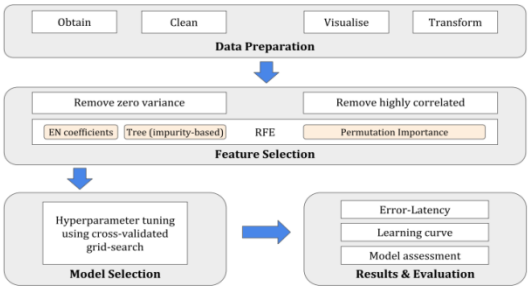

Fig.2: System architecture

MODULES:

To implement aforementioned project we have designed following modules

- Data exploration: using this module we will load data into system

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal www.ijarst.in
ISSN: 2457-0362

- Processing: Using the module we will read data for processing
- Splitting data into train & test: using this module data will be divided into train & test
- Model generation: Build Model- Multitask ElasticNet, Decision tree, Random forest, NLP, Voting Stacking. Algorithms accuracy calculated
- User signup & login: Using this module will get registration and login
- User input: Using this module will give input for prediction
- Prediction: final predicted displayed

4. IMPLEMENTATION

ALGORITHMS:

Multitask ElasticNet: Elastic net linear regression uses the penalties from both the lasso and ridge techniques to regularize regression models. The technique combines both the lasso and ridge regression methods by learning from their shortcomings to improve the regularization of statistical models.

Decision tree: A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

Random forest: A Random Forest Algorithm is a supervised machine learning algorithm which is extremely popular and is used for Classification and Regression problems in Machine Learning. We know that a forest comprises numerous trees, and the more trees more it will be robust.

NLP: NLP algorithms are typically based on machine learning algorithms. Instead of hand-coding large sets of rules, NLP can rely on machine learning to automatically learn these rules by analyzing a set of examples (i.e. a large corpus, like a book, down to a collection of sentences), and making a statistical inference.

Voting Stacking: The fundamental difference between voting and stacking is how the final aggregation is done. In voting, user-specified weights are used to combine the classifiers whereas stacking performs this aggregation by using a blender/meta classifier.

5. EXPERIMENTAL RESULTS

Fig.3: Home screen

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Fig.4: User registration

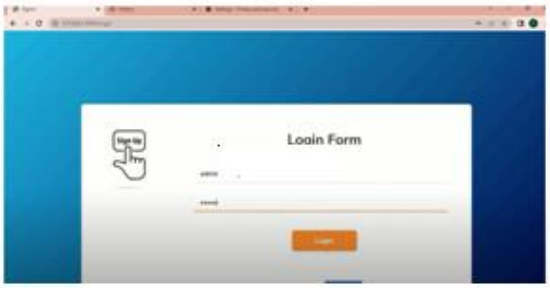

Fig.5: Login page

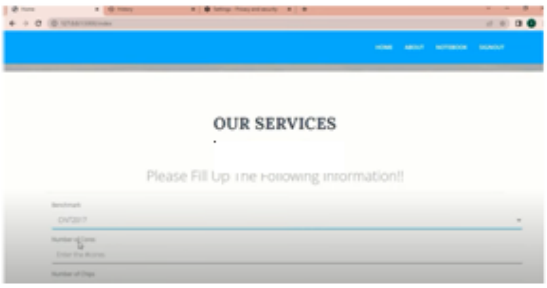

Fig.6: Main page

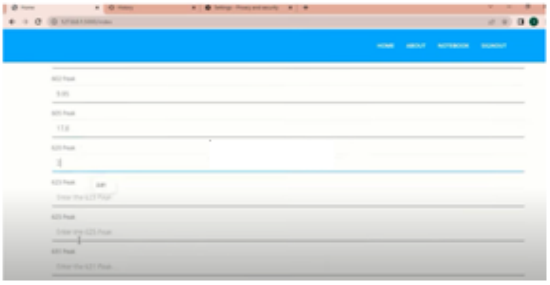

Fig.7: Upload input values

Fig.8: Prediction result

6. CONCLUSION

This study considers whether supervised learning can predict the performance scores of the SPEC benchmarks on parallel systems, without having to actually run the benchmarks. The extensive evaluation has shown that it is possible to accurately predict the performance of parallel and concurrent SPEC CPU2017 benchmarks. Using grid search, we have explored the effect of hyperparameters for each of the above four estimators, and selected the top-10 most accurate models. We have then compared these models together in terms of prediction time (latency) and error. The tree-based models have provided the best results. Also in the RFECV feature selection process, they reach their top performance with smaller feature sets. We have also looked at the learning curves of the topperforming tree-based models to see how accurately we can predict the performance of future systems from the past data. We have shown that for this dataset, even 10% of the past data as the train set can sufficiently predict the future, but 70% or more data can decrease the mean error by a factor of 2 to 3. The current dataset is almost four years old. It would be interesting to see how the figures change as the dataset evolves over time and new generations of systems are added. In the final

step, we have compared the average goodnessof-fit (R2) and MAPE of the final models on the set-aside test set. We have observed that the tree-based models (DT and RF) provide the best R2 and MAPE, both on average and for individual benchmarks. In comparison with the linear models, a probable explanation is that there are still some non-linear relationships that are not captured by the linear models (MT_EN). Compared to the neural networks MLPs, a likely explanation is that tree-based models generally work better when there are different kinds of features involved. Although, neural networks may work better when the number of samples grows beyond a few thousands. So again, it would be interesting to see how these results change as the dataset expands over time. Decision tree and random forest models can keep the average MAPE under 4% with 29 features. Random forests perform better with 10 features though (1.5% < MAPE < 4.5% across the four suites), which make them suitable for building models with smaller numbers of features. However, if interpretability is the main concern, then decision trees will be a better choice. Compared to the previous studies, we have provided more interpretable regression models that can predict the SPEC CPU benchmarks more accurately and offered additional insight into the importance of the hardware and software features used in such models. Using the RFE method, we have found that only a few numbers of hardware and software features (less than 5) are of key importance in our models, and that with just 10 features, we can make highly accurate predictions for this dataset. Our study provides an efficient pipeline for similar performance prediction and evaluation, or design space exploration problems.

REFERENCES

[1] N. Wu and Y. Xie, ‘‘A survey of machine learning for computer architecture and systems,’’ 2021, arXiv:2102.07952.

[2] D. Das, P. Raghavendra, and A. Ramachandran, ‘‘SPECNet: Predicting SPEC scores using deep learning,’’ in Proc. Companion ACM/SPEC Int. Conf. Perform. Eng., Apr. 2018, pp. 29–32.

[3] Y. Wang, V. Lee, G.-Y. Wei, and D. Brooks, ‘‘Predicting new workload or CPU performance by analyzing public datasets,’’ ACM Trans. Archit. Code Optim., vol. 15, no. 4, pp. 1–21, Jan. 2019.

[4] B. Ozisikyilmaz, G. Memik, and A. Choudhary, ‘‘Machine learning models to predict performance of computer system design alternatives,’’ in Proc. 37th Int. Conf. Parallel Process., Sep. 2008, pp. 495–502.

[5] A. Tousi and C. Zhu, ‘‘Arm research starter kit: System modeling using gem5,’’ Arm Res., U.K., Jul. 2017.

[6] J. Bucek, K.-D. Lange, and J. V. Kistowski, ‘‘SPEC CPU2017: Nextgeneration compute benchmark,’’ in Proc. ACM/SPEC Int. Conf. Perform. Eng., 2018, pp. 41–42.

[7] F. Pedregosa, ‘‘Scikit-learn: Machine learning in Python,’’ J. Mach. Learn. Res., vol. 12, pp. 2825–2830, Oct. 2011.

[8] H. Borchani, G. Varando, C. Bielza, and P. Larrañaga, ‘‘A survey on multi-output regression,’’ Data Mining Knowl. Discovery, vol. 5, no. 5, pp. 216–233, 2015.

[9] D. Kocev, S. Džeroski, M. D. White, G. R. Newell, and P. Griffioen, ‘‘Using single- and multi-target regression trees and ensembles to model a

compound index of vegetation condition,'' Ecol. Model., vol. 220, no. 8, pp. 1159–1168, Apr. 2009.

[10] D. Tuia, J. Verrelst, L. Alonso, F. Perez-Cruz, and G. Camps-Valls, ''Multioutput support vector regression for remote sensing biophysical parameter estimation,'' IEEE Geosci. Remote Sens. Lett., vol. 8, no. 4, pp. 804–808, Jul. 2011.