Classification of Computer Hardware and Performance Prediction using Statistical Learning and Neural Networks

Courtney Foots[1], Palash Pal[2], Rituparna Datta[1] and Aviv Segev[1]

[1]Department of Computer Science, University of South Alabama,
Mobile, USA
[2]University Institute of Technology, Burdwan University, India

Abstract

We propose a set of methods to classify vendors based on estimated central processing unit (CPU) performance and predict CPU performance based on hardware components. For vendor classification, we use the highest and lowest estimated performance and frequency of occurrences of each vendor in the dataset to create classification zones. These zones can be used to list vendors who manufacture hardware that satisfy given performance requirements. We use multi-layered neural networks for performance prediction, which accounts for nonlinearity in performance data. Several neural network architectures are analysed in comparison to linear, quadratic, and cubic regression. Experiments show that neural networks can be used to obtain low prediction error and high correlation between predicted and published performance values, while cubic regression can produce higher correlation than neural networks when more data is used for training than testing. The proposed methods can be used to identify suitable hardware replacements.

Keywords

Computer Hardware, Performance Prediction and Classification, Neural Networks, Statistical Learning, Regression

1. Introduction

Computer performance is measured in relation to computational time and valuable work produced and is partly determined by hardware components such as the amount of memory and processor speed [1]. For this study, we are interested in specifically the central processing unit (CPU), which directly affects a computer's performance.

Computer performance prediction can be useful from several perspectives. If an accurate prediction is obtained, it can assist in detecting counterfeit hardware as well as viruses, spyware, Trojans, and other types of malware. Malware and counterfeit components can decrease performance or cause performance instability. There is a plethora of security measures that can be adopted to prevent malicious programs from being downloaded and remove them when they have been downloaded [2, 3]. There are also many novel ways of detecting counterfeit hardware [4, 5]. Thus, methods of detecting malware and counterfeits based on computer performance are useful and part of highly relevant topics in the technology field today.

We can also classify the vendor by the quality of their hardware. Each vendor produces hardware that operates at different standards. These differences could be due to differences in the intellectual property used, as well as the cost point of the hardware. Classifying the hardware based on performance can assist in determining which vendor sells hardware components that will maximize the average performance of the computer. It can also assist in identifying potential hardware replacements that will match the original performance standards.

We propose a set of methods to (1) classify the hardware manufacturer based on estimated CPU performance and (2) predict computer performance based solely on estimated CPU performance using a multi-layered neural network in comparison to regression techniques.

The outline of the rest of this paper is as follows. Section 2 provides a brief overview of related work in the domain of computer performance prediction and computer hardware classification. Section 3 explains the proposed set of methods. Section 4 details our experiments and results. Section 5 presents our conclusions.

2. RELATED WORK

On the topic of classification, Kar et al. [6] proposed a pattern classification model that uses quantitative and qualitative measurements to guide decision making in relation to vendor selection. This tool would assist its user by providing a robust analysis of the supplied collection of vendors so that they may choose the best vendor. In our method, we chose to classify vendors based on quantitative estimated CPU performance data only.

In the discussion of data analysis, Alexander et al. [7] presented a new methodology for analyzing computer performance data using nonlinear time series analysis techniques. The motivation was the concept that computers are deterministic nonlinear dynamic systems. Thus, the previous performance analyses in which computers were considered to be linear and time invariant are not representative of the nature of the actual testing conditions. In our method, we address the same issue of the nonlinearity of our performance data and use neural networks as accommodation.

Hardware performance prediction is a well-studied topic. Lopez et al. [8] explored a way to predict computer performance based on hardware component data without needing simulation. They used a deep learning model to generate a benchmark score for a given hardware configuration, then used multiple neural networks and principal component analysis to predict performance in comparison to the corresponding benchmarks. Neural network and linear regression techniques have been used to predict performance in multiprocessor systems [9]. Similarly, machine learning has been used to predict the performance of multi-threaded applications with various underlying hardware designs [10]. Girard et al. [11] designed a tool to predict the performance of avionic graphic hardware, which is used by engineers to determine the optimal hardware architecture design before manufacturing. Adjacent to the topic of predicting performance, Kang [12] used hardware performance to analyze the microeconomics of buying and leasing computers.

The dataset used in this study has previously been used for detecting scientific anomalies using probability density estimators [13] and fitting linear models in high dimensional spaces [14].

3. PROPOSED SET OF METHODS

3.1. CPU Performance Dataset Description

We aim to classify and predict the performance of CPUs based on a set of ten parameters from an opensource dataset [15]. This dataset contains 209 entries, representing a variety of vendors and models of CPUs. Though the data was donated in 1987, the attributes provided still work well with the scope of our study and are used as test data for the proposed method. The ten parameters of the dataset are listed below:

1. Vendor name
2. Model name
3. Machine cycle time in nanoseconds
4. Minimum main memory in kilobytes
5. Maximum main memory in kilobytes
6. Cache memory in kilobytes
7. Minimum channels in units
8. Maximum channels in units
9. Published relative performance
10. Estimated relative performance from original article [16]

Parameters 1 and 10 are used for vendor classification. Parameters 3 through 9 are used for performance prediction. Parameter 2 is not used in this study. Parameter 10 was calculated using linear regression by Ein-Dor and Feldmesser [16].

3.2. Proposed Classification Method for Hardware Vendors

The highest and lowest estimated performance values are recorded for each vendor, along with the frequency of occurrences of each vendor in the dataset. This information is used to create classification zones. Each zone is labeled with a range of relative performance. The goal is to produce a guide such that given a performance requirement, a list of vendors that manufacture hardware that meet the requirement can be produced.

3.3. Proposed Prediction Method for Hardware Performance

Input parameters 3 through 8 are used to predict the performance of the CPUs. Then, parameter 9 is used with our predicted performance value to calculate the Mean Squared Error (MSE) of the prediction and the correlation between predicted and published performance values. The MSE of the prediction provides insight on the level of accuracy of the prediction in relation to the published performance value. The correlation reflects the percentage of similarity between the predicted and published performance values.

The following is the standard formula used to calculate the MSE for a dataset of n CPUs, where p_i is the predicted performance and p_i' is the published performance:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (p_i - p_i')^2$$

To determine the correlation of the predicted and published performance values, we used the Pearson correlation formula to find the correlation coefficient and the significance level of the

correlation. The correlation coefficient r is calculated as follows, with the values m representing the mean of the predicted and published values:

$$r = \frac{\sum_{i=1}^{n}(p_i - m_{p_i})(p_i' - m_{p_i'})}{\sqrt{\sum_{i=1}^{n}(p_i - m_{p_i})^2 \sum_{i=1}^{n}(p_i' - m_{p_i'})^2}}$$

The significance level of the correlation is ascertained by first calculating the t value as follows:

$$t = \frac{r}{\sqrt{1 - r^2}}\sqrt{n - 2}$$

Then, the corresponding significance level is determined using the t distribution table with a degree of freedom of n - 2. If the significance level is less than 5%, then the correlation between the predicted and published performance values is considered to be significant.

Since the performance of a CPU is affected by other hardware components, there is no perfect or absolute formula to predict its performance. The scatterplot matrix between the inputs and output of our dataset is shown in Figure 1. From this figure, we can see that the relationship between the input variables and the output variable is random and nonlinear. As a result, we use multilayered neural networks, which are suited for random nonlinear input and output relationships.

Specifically, the performance predictions are acquired using various architectures of a multilayered feed forward network with six inputs and one output. When selecting architectures for our tests, we aimed for a variety of hidden layers to determine the level of versatility of the neural network in producing quality results. The inputs to the neural network are the previously discussed parameters. The output is the predicted performance, which is used with parameter 9 to calculate the MSE and correlation values.

We can use regression analysis for the prediction since CPU performance is a continuous measurement. Specifically, we use linear, quadratic, and cubic regression to model the input and output relationship of this dataset and predict performance. The input and output values for each of these are the same used for the neural network.
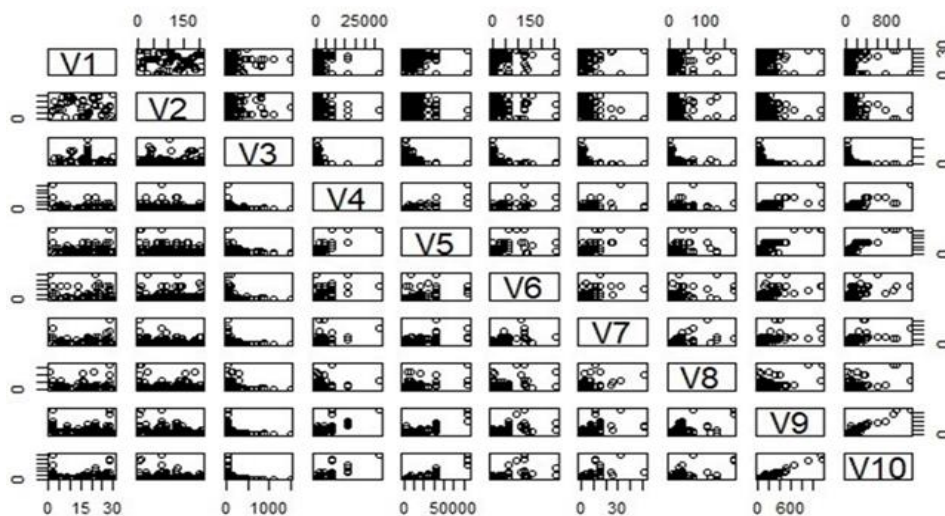

Figure 1. Scatterplot matrix of input and output variables for CPU performance data

4. EXPERIMENTS AND RESULTS

4.1. Vendor Classification Based on Estimated CPU Performance

The classification task is performed with the vendor names and estimated relative performances of each CPU in the dataset. The dataset contains 209 entries with 30 different vendors, out of which the highest and lowest performance values for each vendor as well as the frequency of occurrences of each vendor are tabulated in Table 1.

Table 1: Highest and lowest relative performance values along with frequency of occurrence of each vendor from the CPU performance dataset

Vendor	Highest Performance	Lowest Performance	Frequency
Amdahl	1238	132	9
Sperry	978	24	13
NAS	603	29	19
Siemens	382	19	12
IBM	361	15	32
NCR	281	19	13
Adviser	199	199	1
Honeywell	181	20	13
Gould	157	75	3
CDC	138	23	9
IPL	128	30	6
Burroughs	124	22	8
BASF	117	70	2
Magnuson	88	37	6
Cambex	74	30	5
DG	72	19	7
Nixdorf	67	21	3
Perkin-Elmer	64	24	3
BTI	64	15	2
HP	54	18	7
DEC	54	18	6
Prime	53	20	5
Harris	53	18	7
Wang	47	25	2
Stratus	41	41	1
Formation	34	34	5
Microdata	33	33	1
C.R.D	28	21	6
Apollo	24	23	2
Four-Phase	19	19	1

The classification result is shown in Figure 2. According to the results, the vendors can be classified into five zones. Each zone represents a performance standard, with Zone I being the lowest relative performance of 200 or less, and Zone V being the highest relative performance of

1000 or more. If the desired relative performance is less than 200, any vendor can be chosen. If the performance requirement is from 200 – 400, any one of NCR, IBM, Siemens, NAS, Sperry, or Amdahl can be chosen. The vendors NAS, Sperry, or Amdahl can be chosen for a performance requirement from 400 – 600. The vendors Sperry or Amdahl can be chosen for a performance requirement from 600 – 1000. Last, only Amdahl can be chosen for performance requirements more than 1000.
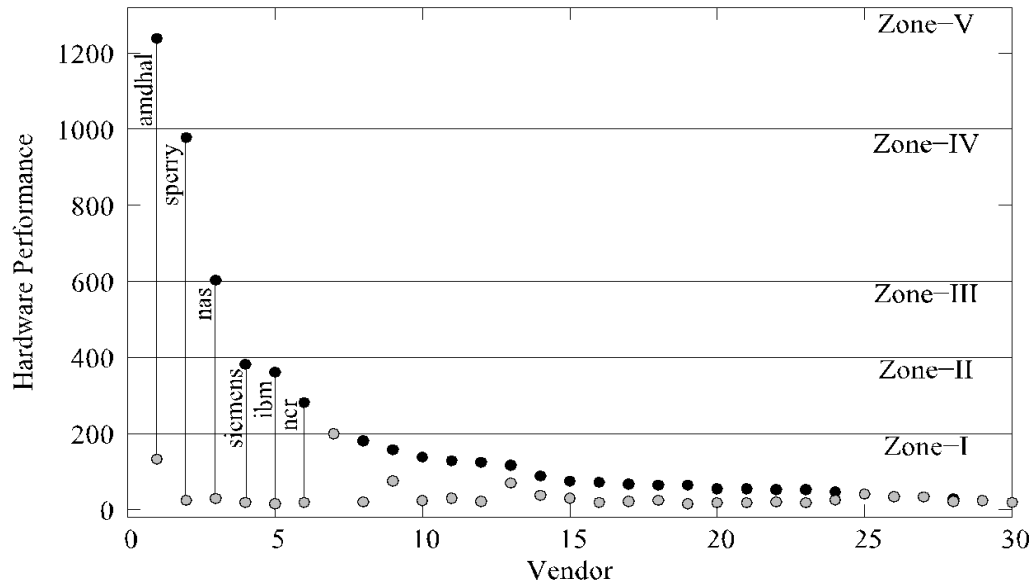

Figure 2. Classification of vendors based on estimated relative performance for CPU performance prediction

4.2. Computer Performance Prediction Based on Hardware Components

After scaling the data using Min-Max scaling, we construct several multilayered neural network architectures with various numbers of hidden layers. In Tables 2 through 5, the training-testing percentages represent the proportion of the dataset that was used for training and testing, respectively. The sequences of architecture values represent the number of neurons in each layer of the neural network. The scaled MSE and correlation of the predictions are calculated for each neural network architecture at each training-testing ratio. We also calculate the MSE and correlation of the predictions using linear, quadratic, and cubic regression analysis. All correlation coefficients have a significance value less than 5%, except for quadratic regression at 62.5% - 37.5% training-testing ratio. Therefore, the correlation coefficients between predicted and published performances for this study do have significance.

For all training-testing ratios, the lowest MSE values are produced by the neural network. For training-testing ratios 62.5% - 37.5% and 65% - 35%, the highest correlation values are produced by the neural network. For training-testing ratio 67.25% - 32.75%, the highest correlation value is produced by cubic regression, with the neural network outperforming linear and quadratic regression. For training-testing ratio 70% - 30%, the highest correlation values are produced by cubic and quadratic regression, with the neural network outperforming only linear regression.

Table 2: The MSE and predicted-published performance correlation for CPU performance prediction with 62.5% - 37.5% training-testing ratio

Training-Testing	Method	Architecture	Scaled	Correlation	Significance
62.5% - 37.5%	Neural Network	$6-3-1$	0.00357	0.913	2.2 e-16
		$6-4-2-1$	0.00387	0.909	2.2 e-16
		$6-4-3-2-1$	**0.00307**	**0.924**	2.2 e-16
		$6-5-4-3-2-1$	0.00369	0.914	2.2 e-16
	Regression	Linear	**0.00629**	0.848	2.2 e-16
		Quadratic	0.02555	0.136	0.2326
		Cubic	0.01549	**0.898**	2.2 e-16

Table 3: The MSE and predicted-published performance correlation for CPU performance prediction with 65% - 35% training-testing ratio

Training-Testing	Method	Architecture	Scaled	Correlation	Significance
65% - 35%	Neural Network	$6-3-1$	**0.00190**	0.956	2.2 e-16
		$6-4-2-1$	0.00209	0.951	2.2 e-16
		$6-4-3-2-1$	0.00359	0.915	2.2 e-16
		$6-5-4-3-2-1$	0.00197	**0.958**	2.2 e-16
	Regression	Linear	0.00470	0.884	2.2 e-16
		Quadratic	0.00260	0.940	2.2 e-16
		Cubic	**0.00242**	**0.944**	2.2 e-16

Table 4: The MSE and predicted-published performance correlation for CPU performance prediction with 67.25% - 32.75% training-testing ratio

Training-Testing	Method	Architecture	Scaled	Correlation	Significance
67.25% - 32.75%	Neural Network	$6-3-1$	0.00284	0.934	2.2 e-16
		$6-4-2-1$	0.00223	**0.954**	2.2 e-16
		$6-4-3-2-1$	0.00342	0.920	2.2 e-16
		$6-5-4-3-2-1$	**0.00220**	0.952	2.2 e-16
	Regression	Linear	**0.00500**	0.884	2.2 e-16
		Quadratic	0.00757	0.910	2.2 e-16
		Cubic	0.00685	**0.961**	2.2 e-16

Table 5: The MSE and predicted-published performance correlation for CPU performance prediction with 70% - 30% training-testing ratio

Training-Testing	Method	Architecture	Scaled	Correlation	Significance
70% - 30%	Neural Network	$6-3-1$	0.00351	**0.898**	2.2 e-16
		$6-4-2-1$	**0.00332**	0.886	2.2 e-16
		$6-4-3-2-1$	0.00414	0.880	2.2 e-16
		$6-5-4-3-2-1$	0.00348	0.867	2.2 e-16
	Regression	Linear	**0.00355**	0.850	2.2 e-16
		Quadratic	0.00386	0.922	2.2 e-16
		Cubic	0.01415	**0.936**	2.2 e-16

The best performing methods, architectures, and training-testing ratios are compared in Figure 3 with respect to lowest MSE and in Figure 4 with respect to highest correlation. It is clear from Figure 3 that the lowest MSE overall is obtained using architecture 6-3-1 with a training-testing ratio of 65% - 35%. Figure 4 shows that the highest correlation overall is obtained using cubic regression with a 67.25% - 32.75% training-testing ratio. A plot of the published vs. predicted CPU performance by the neural network with 65% - 25% training-testing ratio and 6-3-1 architecture is shown in Figure 5, and the best performing neural network architecture is shown in Figure 6. There appears to be no correlation between the number of hidden layers in the architecture of a neural network and the relative performance of the neural network.
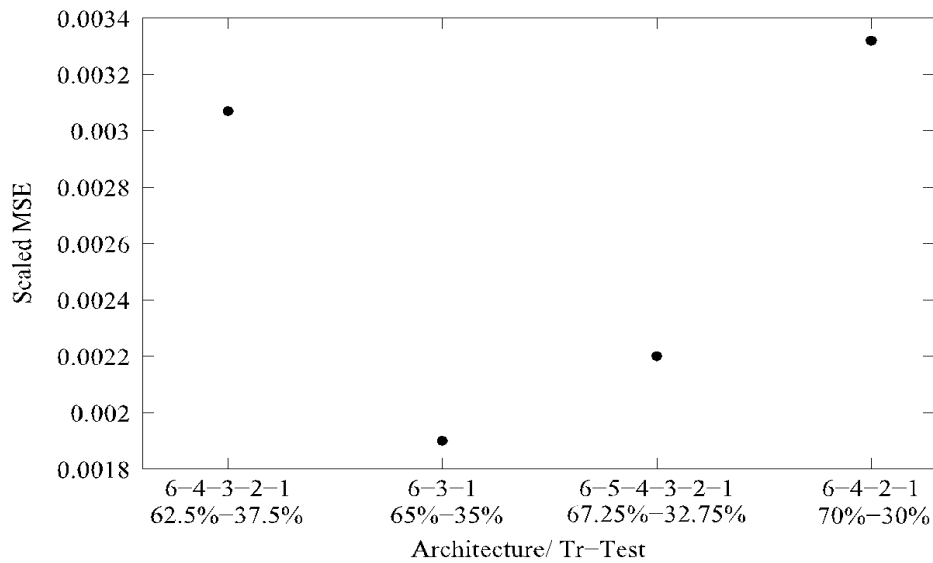

Figure 3. Comparison of architectures along with training testing ratio with respect to scaled Mean Squared Error for CPU performance prediction
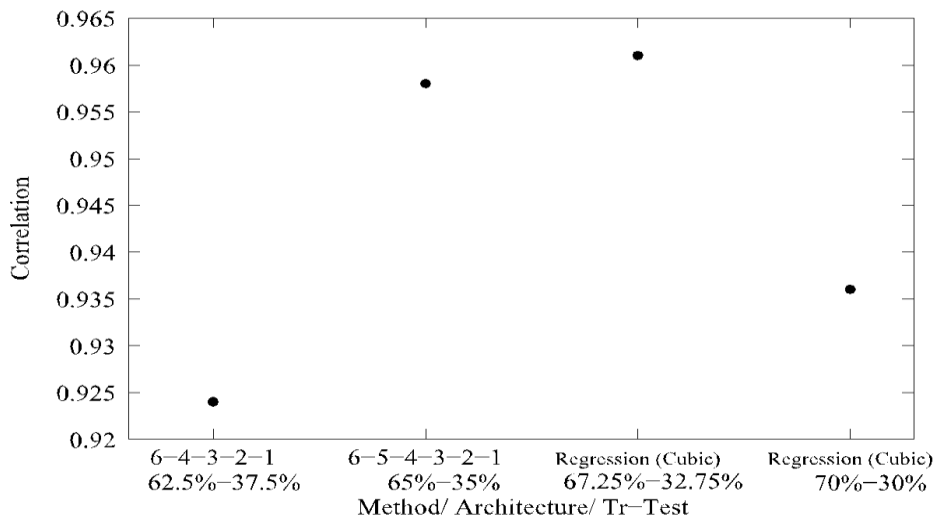

Figure 4. Comparison of methods and architectures along with training testing ratio with respect to correlation for CPU performance prediction
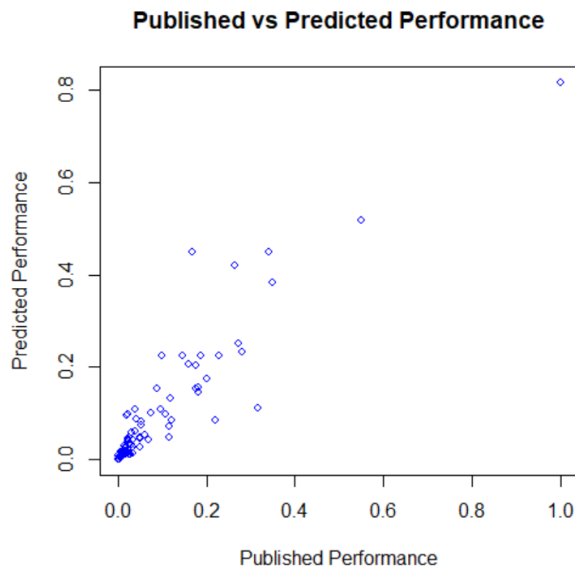
Figure 5. Plot of the published vs. predicted CPU performance for neural network with 65% - 25% training-testing ratio and 6-3-1 architecture
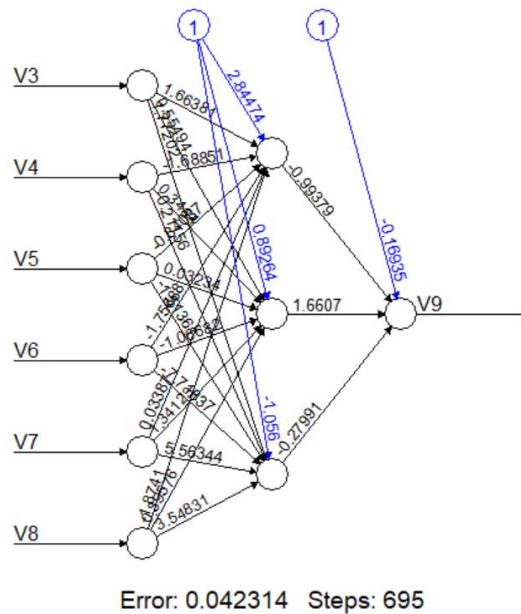

Error: 0.042314   Steps: 695

Figure 6. Graphical representation of the 6-3-1 neural network architecture

5. CONCLUSIONS

In this paper, both classification and prediction tasks are performed to analyze the performance of CPUs documented in our test dataset. The classification study shows that 30 vendors can be successfully classified into 5 performance zones. Each zone provides information about the relative performance capabilities of the vendors' hardware. Performance prediction is generated using neural network and regression techniques which accommodate the random, nonlinear relationship between input and output variables. Among all prediction results, the neural network with 65% training - 35% testing ratio and 6-3-1 architecture performs the best in terms of having

the lowest scaled MSE. However, the performance of cubic regression with 67.25% training - 32.75% testing ratio is found to be best in terms of the highest correlation. The numerous experiments with varying architectures and training-testing ratios show that the obtained results are robust. The results from our Pearson correlation analysis show that the correlations between the predicted and published performance values are significant.

The results from our performance prediction study show that neural networks can be used to obtain lower prediction error and often higher significant correlation between predicted and published values. However, cubic regression may have better predicting capabilities than a neural network when a higher percentage of the data is used for training. While this does reiterate our initial observation that there is no perfect or absolute method of predicting CPU performance, this study can be extended to analyze how hidden layers in the architecture of the tested neural networks affect their predicting capabilities. The prediction method can be also used on current hardware performance data to determine if neural networks outperform regression for a more robust range of experimental structures.

Our classification result shows that other than Zone-V, a given required performance can be obtained by more than one vendor. While this result does show that suitable replacement hardware can be found using this method, it also implies that hardware configuration can be copied or tampered with while still having nearly the same performance as the original configuration. To alleviate this drawback, the work will be extended to detect counterfeit hardware by a more thorough analysis and comparison of computer hardware performance

ACKNOWLEDGEMENTS

REFERENCES

[1]   Lilja, D. (2005) *Measuring Computer Performance: A Practitioner's Guide*, University of Minnesota, Cambridge University Press.

[2]   Aslan, O. & R. Samet (2020) "A Comprehensive Review on Malware Detection Approaches" *IEEE Access*, Vol. 8, pp 6249 - 6271.

[3]   Bakhshinejad, N. & A. Hamzeh (2020) "Parallel-CNN Network for Malware Detection," *IEEE Information Security*, Vol. 14, Issue 2, pp 210 - 219.

[4]   Wang, X., Y. Han, & M. Tehranipoor (2019) "System-Level Counterfeit Detection Using On-Chip Ring Oscillator Array," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 27, Issue 12, pp 2884 – 2896.

[5]   Chattopadhyay, S., P. Kumari, B. Ray, R. Chakraborty (2019) "Machine Learning Assisted Accurate Estimation of Usage Duration and Manufacturer for Recycled and Counterfeit Flash Memory Detection," *2019 IEEE 28th Asian Test Symposium (ATS)*, Kolkata, India, pp. 49-495.

[6]   Kar, A., A. Pani, B. Mangaraj, & S. De (2011) "A Soft Classification Model for Vendor Selection," *International Journal for Information and Education Technology*, Vol. 1, No. 4, pp 268 - 272.

[7]   Alexander, Z., T. Mytkowicz, A. Diwan, & E. Bradley (2010) "Measurement and Dynamical Analysis of Computer Performance Data" *Springer, Berlin, Heidelberg*, Lecture Notes in Computer Science, Vol. 6065, pp 18-29.

[8]   Lopez, L., M. Guynn, & M. Lu (2018) "Predicting Computer Performance Based on Hardware Configuration Using Multiple Neural Networks," *IEEE,* 17th IEEE International Conference on Machine Learning and Applications (ICMLA).

[9]   Ozisikyilmaz, B., G. Memik, & A. Choudhary (2008) "Machine Learning Models to Predict Performance of Computer System Design Alternatives," *IEEE*, 37th IEEE International Conference on Parallel Processing.

[10]   Agarwal, N., T. Jain, & M. Zahran (2019) "Performance Prediction for Multi-threaded Applications," *International Workshop on AI-assisted Design for Architecture*.

[11]   Girard, S., V. Legault, G. Bois, & J. Boland (2019) "Avionics Graphics Hardware Performance Prediction with Machine Learning," *Scientific Programming*.

[12]   Kang, Y. (1989) "Computer hardware performance: production and cost function analyses," *Communications of the ACM*, Vol. 32, No. 5, pp 586 –593.

[13]   Pelleg, D. (2004) "Scalable and Practical Probability Density Estimators for Scientific Anomaly Detection," School of Computer Science, Carnegie Mellon University.

[14]   Wang, Y. (2000) "A New Approach to Fitting Linear Models in High Dimensional Spaces," Department of Statistics, University of Auckland.

[17]   Ein-Dor, P. & J. Feldmesser, Donor: W. Aha (1987) *Computer Hardware Data Set*, UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/Computer+Hardware.

[16]   Ein-Dor, P. & J. Feldmesser (1987) "Attributes of the Performance of Central Processing Units: A Relative Performance Prediction Model," *Communications of ACM*, Vol. 30, pp 308-317.

AUTHORS

Courtney Foots is researching as an undergraduate student at the University of South Alabama, studying computer science and mathematics. Her research interests include counterfeit hardware detection, applications of artificial intelligence, and optimization.

Palash Pal received a bachelor's degree in Technology from University Institute of Technology, Burdwan University, West Bengal, India.

Rituparna Datta is working as a Computer Research Associate-II in the Department of Computer Science, University of South Alabama. Prior to that, he was an Operations Research Scientist in Boeing Research & Technology (BR&T), BOEING, Bangalore.

Aviv Segev is working as an Associate Professor at the Department of Computer Science, University of South Alabama. His research interest is looking for the DNA of knowledge, an underlying structure common to all knowledge, through analysis of knowledge models in natural sciences, knowledge processing in natural and artificial neural networks, and knowledge mapping between different knowledge domains.