Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 179 (2021) 515–523

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on Computer Science and Computational Intelligence 2020

# Performance of Information Technology Infrastructure Prediction using Machine Learning

Ignatius Rahardjo Heruwidagdo[a], Suharjito[a,b], Novita Hanafiah[b,]*, Yanto Setiawan[b]

[a]*Computer Science Department, Binus Graduate Program-Master in Computer Science, Bina Nusantara University, Indonesia*
[b]*Computer Science Department, BINUS Online Learning, Bina Nusantara University, Jakarta, Indonesia 11480*

**Abstract**

Resource management is always an important issue related to good governance decision making. One of the common problem faced in managing IT Infrastructure is about allocating server resources to improve the performance. In this study we use a machine learning approach to make predictions about the performance of information technology infrastructure. The experiment took data from several servers in a company to be tested. The performance measure of resources used in this study are CPU Performance, Disk performance, Memory capacity, and Network performance. Several algorithms and machine learning methods are tested, such as Linear Regression, kNN, SVR, Decision Tree and Random Forest, to find the best model fit for the servers. The comparison result shows that Linear regression and kNN perform well in predicting the network performance in those three servers.

*Keywords:* log performance, machine learning, kNN, IT infrastructure, performance prediction;

## 1. Introduction

A common problem faced in managing Information Technology (IT) infrastructure is poorly managed resources, especially how governance allocates server resources. Resource management is always an important issue because

* Corresponding author. Tel.: +62 858 8378 7771.
*E-mail address:* novita.hanafiah@binus.ac.id

10.1016/j.procs.2021.01.035

good governance is very helpful in solving problems and helping in decision making. Another problem in managing IT infrastructure is the absence of models or forms used as scientific references or standards in determining the design and improving the performance of IT infrastructure facilities, so that the system is easily managed and developed in support of business processes. The choice of models determines the direction of different strategies and approaches. The selection of different strategies, of course, also results in various approaches to cost savings and efficiency [1].

Machine learning is used in this study as a tool in making approaches in determining existing infrastructure resource models. The first step is to analyze, and process data sourced from the server performance log, then create a model by determining input and output variables. Furthermore, the data processed with several algorithms in machine learning to obtain mathematical models of the use of server infrastructure resources. By modeling server that is used every day, it can be used to design and improve the performance of IT infrastructure [2]. Data taken from the server performance log in this study consists of several variables such as: CPU usage, disk performance, memory performance and network performance. We experiment with several methods, such as: Linear Regression, SVR, kNN, Decision Tree and Random Forest. The infrastructure referred to in this study, can be in the form of hardware (physical servers) or software (virtual servers) in a network.

Some Research in Grid and Cloud models for resources management are discussed in section 2. Section 3 present the methodology of our research and section 4 shows the experiment result which is done by analyzing the performance of server infrastructure on a running system. The conclusion is presented in section 5.

## 2. Related Work

### *2.1. Machine Learning Methods*

Machine Learning can be used to produce predictions and improve systems, by making more accurate decisions from available information (execution, resources, and requirements). Machine Learning uses techniques based on past state information records to then create an appropriate model of the general situation, further continuing the anomaly system to produce results and system evaluations. In other words, the use of Machine Learning in IT infrastructure governance can explore knowledge directly on the behavior of systems that are running [2]. In general, the way Machine Learning works is by processing a series of data called data sets, which originate from a system by determining system values, determining which attributes and which are the responses, then making a model based on those values, so that when there is data new, the expected value will be in accordance with the expectations of the model obtained.

Supervised Learning is one of the techniques in Machine Learning that directly learns from the operational data of a system, which allows the prediction of the system. Data for the purposes of prediction itself, consisting of various elements of information that can come from various sources. Model selection helps in making predictions or estimates of information and can show how the relationship between observed data with the state of the system that is running. Research conducted by Josep LL. Berral et al. [2] adopted for our research, which is based on a grid model. The grid model existed before the cloud model. Some researches based on other grid and cloud models are shown in Table 1.

Table 1. Research in Grid and Cloud models for resources management

| No. | Author | Research Topic | Methods |
|---|---|---|---|
| 1 | Roy, C., Moitra, S., Das, M., Srinivasan, S., & Malhotra, R. [3] | IT Infrastructure Management | Support Vector Machine, Random Forest |
| 2 | Gao, J. [4] | Data Center Optimization | Neural Network |
| 3 | Huang, C. J., Wang, Y. W., Guan, C. T., Chen, H. M., & Jian, J. J. [1] | Resource Allocation | Support Vector Regression,<br>Genetic Algorithm |
| 4 | Fang, Z. [5] | Resource Management | Linear Regression, Decision Tree |

| | | | |
|---|---|---|---|
| 5 | Alonso, J., Torres, J., & Gavalda, R. [6] | Memory, CPU Monitoring and Prediction for Web Application | Linear Regression, Decision Tree |
| 6 | Zhang, Q., Cherkasova, L., Mi, N., & Smirni, E. [7] | Prediction for Capacity Planning | Regression |
| 7 | Kapadia, N. H., Fortes, J. A., & Brodley, C. E. [8] | Performance Prediction | Nearest-Neighbour, Polynomial Linear Regression |
| 8 | Verma, N., & Sharma, A. [9] | Virtual Machine workload prediction | Linear Model, Conditional Inference Tree, Boosted Tree |
| 9 | Tseng, Fan-Hsun, et al. [10] | Forecast the resource utilization and energy consumption in cloud data center | Multiobjective Genetic Algorithm |
| 10 | Shyam, G. K., & Manvi, S. S. [11] | Virtual resource prediction in cloud | Bayesian Model |

On the other hand, solving problems with unsupervised learning includes using Cluster Analysis [12]. The aim is to make a grouping or grouping of a set of objects into smaller groups (subsets) by obtaining a degree of similarity (or difference) between individual objects in a cluster. Some techniques that can be used in clustering include: Proximity Matrix, Dissimilarity Based on Attributes, and Dissimilarity of Objects.

### *2.2. Attribute and Target Determination*

According to Bowles, M. [13], the Machine Learning algorithm starts from collecting a series of data, looking at existing data, determining which components will determine predictions. These components can be divided into attributes and target. Attributes are variables used as inputs in making predictions, also known as some other terms, such as predictor, feature, independent variables, input. Target is desired thing as a prediction, also known as some other terms, such as outcomes, labels, dependent variables, response. In other words, the attribute is needed to predict the target. After determining which data components become attributes and which are the target components, then the predictive model is determined based on the type of attribute and target. Attributes can be divided into attributes with numeric variables and attributes with categorical variables or factors. Target components are divided into numeric targets and categorical targets.

### *2.3. Bias – Variance Trade off*

Bias and variance are things to consider in calculating predictions using machine learning. As mentioned above, the prediction equation in machine learning is $Y = f(X) + \varepsilon$ . [12] If $E(\varepsilon) = 0$ and $\mathrm{Var}(\varepsilon) = \tau\, \varepsilon^2$, then we get the prediction error equation from the regression approach with $\hat{f}(X)$ at an input point $X = x_0$, using squared-error loss:

$$\begin{aligned} \mathrm{Err}(x0) &= E[(Y - \hat{f}(x0))^2 \mid X = x0] \\ &= \tau\, \varepsilon^2 + [E \hat{f}(x0) - f(x0)]^2 + E[\hat{f}(x0) - E \hat{f}(x0)]^2 \\ &= \tau\, \varepsilon^2 + \mathrm{Bias}^2(\hat{f}(x0)) + \mathrm{Var}(\hat{f}(x0)) \\ &= \text{Irreducible Error} + \mathrm{Bias}^2 + \text{Variance} \end{aligned} \tag{1}$$

Irreducible Error is a target variance around the average value of f (x0), and is an unavoidable error value, though as good as any we do estimates of $f(x_0)$, except $\tau\, \varepsilon^2 = 0$. $\mathrm{Bias}^2$, squared bias, is an average estimation value different from the actual average value. Variance is the expected squared deviation value $\hat{f}(x_0)$ around its mean value. In general, the more complex the modeling is $\hat{f}$, then the lower the bias (squared), but the higher the variance.

### 2.4. Modeling

Linear regression is a natural choice for consideration. Linear equations with many attributes (features), can be written as follows:

$$y = w_0 + w_1 a_1 + w_2 a_2 + \ldots + w_k a_k \tag{2}$$

Where *y* is prediction, *a* is feature, and *w* is coefficient. Coefficient value obtained from training data.
The above equation, when written in vector form for n-observation, becomes Y = AW + $\mathcal{E}$ where *Y* is dependent variable, *A* is independent variable, *W* is regression coefficient and $\mathcal{E}$ is error. The prediction value is defined as (3).

$$w_0 a_0{}^{(1)} + w_1 a_1{}^{(1)} + w_2 a_2{}^{(1)} + \ldots + w_k a_k{}^{(1)} = \sum_{j=0}^{k} w_j a_j^{(i)} \tag{3}$$

While the difference between the actual value (data test) and the predicted value is:

$$\sum_{i=1}^{n} (y^{(i)} - \sum_{j=0}^{k} w_j a_j^{(i)})^2 \tag{4}$$

The equation in parentheses is the difference between the actual value and the predicted value for the *i*-th observation. The difference between the actual value and the predicted value is taken by the smallest value, as the score value for the linear regression model. The alternative measure of fit, using R$^2$, which is the proportion of Variance that has a value between 0 and 1 [14]. The formula for R$^2$, is written as follows:

$$R^2 = \frac{\text{TSS} - \text{RSS}}{\text{TSS}} = 1 - \frac{\text{RSS}}{\text{TSS}} \tag{5}$$

And TTS = $\sum (y_i - \bar{y})^2$ (total sum of squares, TSS) (6)

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \text{(residual sum of squares, RSS)} \tag{7}$$

R$^2$ measures the proportion of variation in determining the value of Y by using X. The value of R$^2$ close to 1 means the appropriate regression model is used. While value of R$^2$ close to 0 means the Regression model is not appropriate.

## 3. Methodology

Our research method uses supervised learning approaches which are linear regression and classification methods. Linear Regression produces output in the form of real data to get real results and makes predictions of results about how real the optimal value of the application is needed, so that server performance can be calculated properly. Whereas classification produces output in the form of discrete data to classify - divide into several classes or groups - to the data obtained. In addition, we also use Support Vector Regression (SVR), Neighbor K-Nearest (KNN), Decision Tree, and Random Forest in our experiment. The framework in this study is shown in Figure 1.

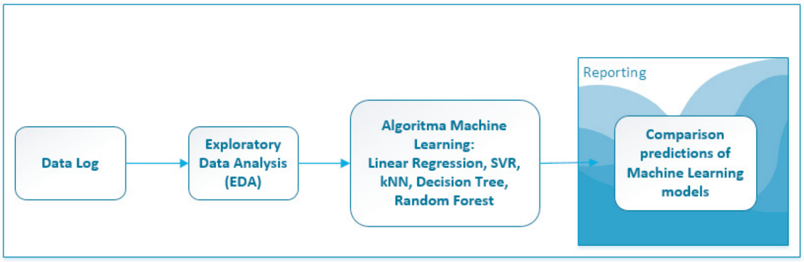

Data Log
Exploratory Data Analysis (EDA)
Algoritma Machine Learning:
Linear Regression, SVR, kNN, Decision Tree, Random Forest
Reporting
Comparison predictions of Machine Learning models

Fig. 1. Research Framework

Measures of resource performance that is used in this study are CPU performance (MHz), disk performance (KBps), memory capacity (KB), and network performance (KBps) when running applications, such as research conducted by Chenn-Jung Huang, et al [1]. The computation in their experiment used Cloud, while our research is done for in-company networks (LAN). Figure 2 shows the stages of this study.

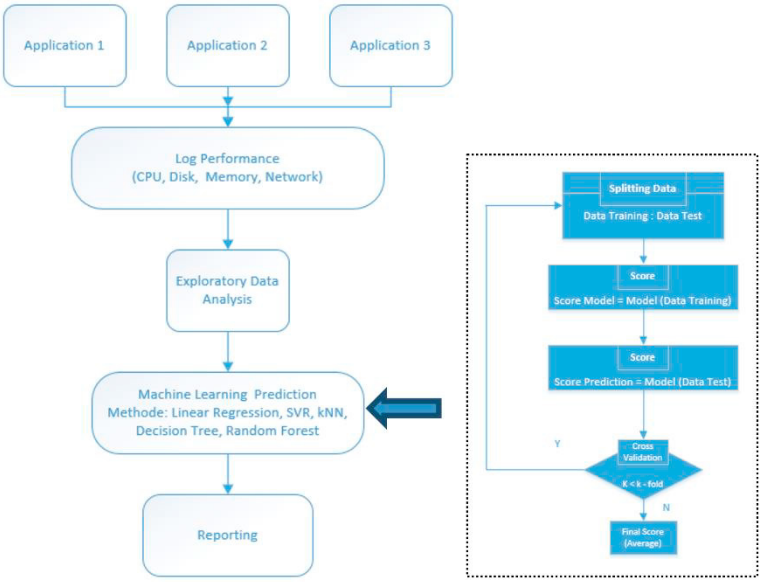

Application 1
Application 2
Application 3
Log Performance
(CPU, Disk, Memory, Network)
Exploratory Data Analysis
Machine Learning Prediction
Methode: Linear Regression, SVR, kNN, Decision Tree, Random Forest
Reporting
Splitting Data
Data Training : Data Test
Score
Score Model = Model (Data Training)
Score
Score Prediction = Model (Data Test)
Y
Cross Validation
K < k - fold
N
Final Score (Average)

Fig. 2. Research Stages

The first step is to take a log file (with parameters: processor, memory, disk, and network performance) for the application system / business process from Application 1, Application 2 and Application 3. The log files are combined and transformed into a table format. There are 357 rows of observational data with 4 colums: CPU Usage (MHz), Disk Performance (Usage, KBps), Memory Performance (Consumed, KB), and Network Performance (Usage, KBps). Then an Exploration Data Analysis (EDA) is carried out. The data is normalized by dividing each column data with the maximum value in each column. Normalization needs to be done so that no column dominates other columns because the numeric numbers are too high against the other columns. Next the process is continued with Information Technology infrastructure prediction performance (using Python-based programming language). At this stage there are several things that are done.

- Dividing data that has been changed in table format into training data and test data.
- Modeling for single and multi-feature features, starting with the Linear regression algorithm.

- Feature Selection: Selection of features to show how the relationship between independent variables with the dependent variable (using Lasso Regularization).
- Evaluation models: each model for the learning process of single features and multi-features for comparison of data split between training data and test data with three variations (60:40, 70:30, 80:20). Then a score model is determined (for train data and test data: large scores), RMSE scores, and linear model equations.
- Model improvement: done by applying cross-validation where we divide training data into training data and validating data (for k-fold = 3, 5 and 10). Then a score model is determined (for train data and test data: a large score is taken), an RMSE score, and a predictive accuracy of the cross-validation results.
- Comparing the results of linear regression with other learning algorithms, such as: KNN, SVR, Decision Tree and Random Forest.

The final step is to make a report (for Decision Making) by displaying a comparison chart between the predicted curve and the actual value.

## 4. Experiment Result

We experimented with 3 applications. The relationship between the input and output variables is illustrated using a scatter plot, as well as to see the observed distribution between them (in two dimensions). The data distribution for application 1, 2 and 3 are shown in Figure 3.

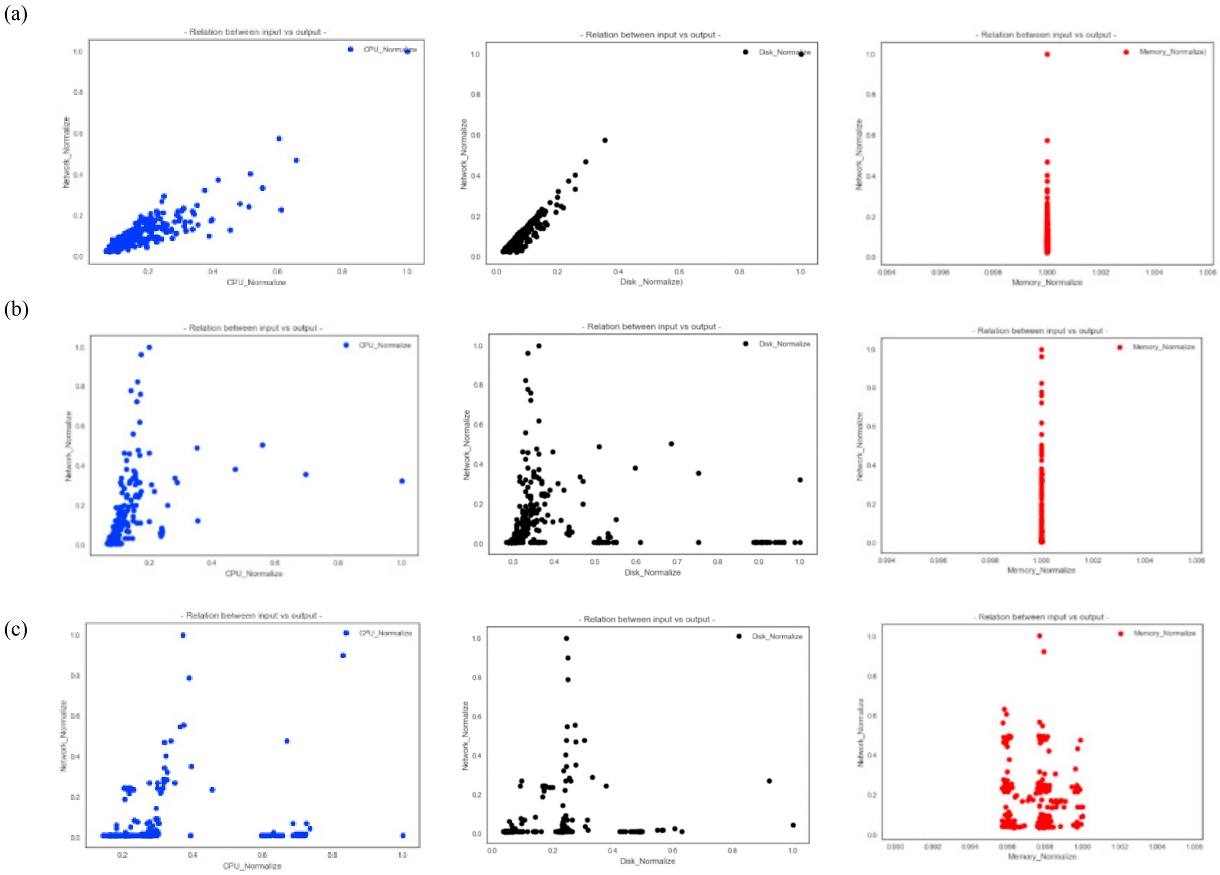


Fig. 3. Data Distribution for CPU usage, Disk and Memory (a) Application 1; (b) Application 2; (c) Application 3

Splitting data is done by several combinations, 60:40, 70:30 and 80:20. Then the best score is chosen. Table 2 compares the results of variance scores (R2) and bias scores (RMSE) for combinations of various data split variations. The table shows that the best value variance score is obtained almost for all split data but the selected score variance (R2) is 0.85986 with a bias score (RMSE) of 0.02265 as model choice (at 70:30 split data) for application 1. The best score variance (R2) from application 2 was chosen in the 60:40 split data, with the smallest bias (RMSE), among others, 0.13350. On the other hand, the table shows that the bad model scores are obtained for all split data in application 3, whether at 60:40, 70:30, or 80:20, with a low RMSE value. But the best variant value ($R^2$) is chosen at 70:30 variant, with the smallest bias (RMSE), among others, 0.12283.

Table 2. Split Data Score Model for Application server 1, 2 and 3

| Scoring Function | Application 1 | | | Application 2 | | | Application 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| | 60:40 | 70:30 | 80:20 | 60:40 | 70:30 | 80:20 | 60:40 | 70:30 | 80:20 |
| Score Variance (R2) | 0.85865 | **0.85986** | 0.84738 | **0.27102** | 0.25020 | 0.24668 | 0.00333 | 0.00846 | 0.00125 |
| Score Bias (RMSE) | 0.02189 | **0.02265** | 0.02016 | **0.13350** | 0.13445 | 0.14823 | 0.11257 | 0.12283 | 0.14667 |

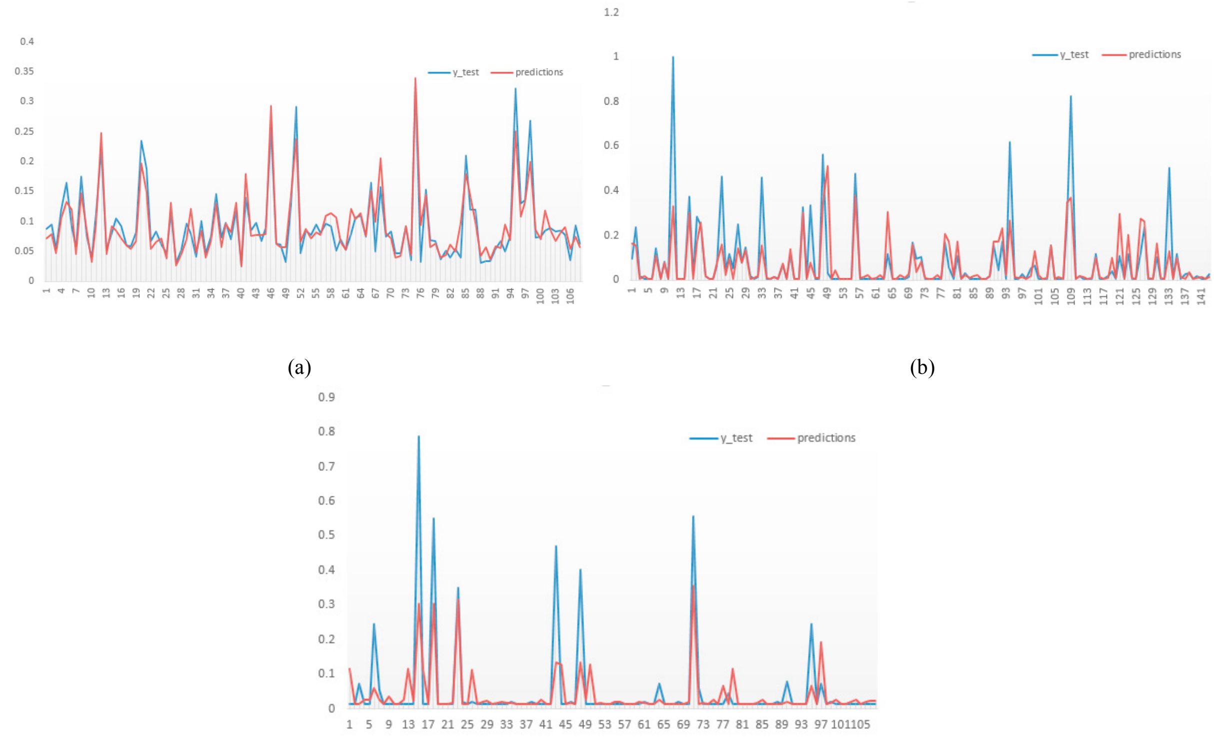
Furthermore, the accuracy of the model can be improved, by splitting training data randomly, using the Cross-Validation method. In this study, several combinations are used to determining the k-fold value: cv = 3, cv = 5 and cv = 10. From table 2 we get that the best splitting combination test data is 70: 30 for application 1 & 3, and 60:40 for application 2, so we experimented the test model for three k-fold variations (cv = 3, cv = 5, cv = 10) in that combination. The value of the model score is corrected by using the cross-validation method, and the highest score of the model score is about 85% for application 1 (score variance (R2) = 0.852, for cv = 3) and 25% for application 2 (score variance, $R^2$ = 0.253259064, for cv = 10). High variance and RMSE values (after valid cross validation) still applicable in application 3. So the variance and RMSE values used in application 3 after cross validation: the variance score ($R^2$): 0.01963 and the bias score (RMSE): 0, 12214 (at cv = 5).

Table 3 shows the comparison of data processing results from Application1 with some features for various other machine learning models, such as kNN, SVR, Decision Tree and Random Forest. The table shows that the model for Application 1 has a good score with linear regression modeling with the largest variance score (R2): 0.852600227 and the smallest bias score (RMSE): 0.023234272. The results show that the largest score is also obtained for the Linear Regression model compared to other algorithm models for Application 1. Meanwhile the model for Application 2 has a good score value with kNN modeling (with variant score 0.522948878 and bias score RMSE: 0.107998185). Similar to application 2, the model for application 3 has a good score with kNN modeling (with variant score: 0.574224537 and bias score: 0.080493473).

Table 3. Comparison with all algorithms for 3 applications

| Application-n | Scoring function | Linear Regression | KNN | SVR | Decision Tree | Random Forest |
|---|---|---|---|---|---|---|
| Application 1 | Score Variance ($R^2$) | **0.852600227** | 0.7233123 | 0.1294902 | 0.675526677 | 0.644596589 |
| | Score Bias (RMSE) | **0.023234272** | 0.03183285 | 0.06431643 | 0.034472292 | 0.036077917 |
| Application 2 | Score Variance ($R^2$) | 0.253259064 | **0.522948878** | 0.176564862 | 0.412739182 | 0.43640361 |
| | Score Bias (RMSE) | 0.135119795 | **0.107998185** | 0.141888989 | 0.119825572 | 0.117386485 |
| Application 3 | Score Variance ($R^2$) | 0.019631836 | **0.574224537** | 0.263672213 | 0.175422563 | 0.158360278 |
| | Score Bias (RMSE) | 0.122141979 | **0.080493473** | 0.138671614 | -1.02222835558 | 0.11317056 |

Figure 4 shows the prediction of network usage (resources: CPU, Disk, Memory) from application servers 1, 2 and 3. Comparison between test data results and prediction results for each application server are presented. Figure 4a presents the results of comparisons for Application 1 using the Linear Regression model and we can see the trend matches between the (actual) test data with the predicted trend. Similar to Application 1, The result of other servers (application 2 and 3) that use the kNN model show a harmonious trend between the (actual) test data and the predicted trend.



(a)

(b)

(c)

Fig. 4. Actual Vs Prediction result (a) Application 1; (b) Application 2; (c) Application 3

## 5. Conclusion

This research is successful in creating a prediction model of IT infrastructure performance using server log data, with CPU usage, disk performance, memory performance, and network performance as variables. The experiment is done by analyzing the performance of server infrastructure on a running system, in running applications that support business processes (Application1, Application2, Application3), which bases data on server performance logs in each application. We can also conclude that it is important to test the model with the cross-validation step. It can be seen that the model scores (for variant scores ($R^2$) and bias (RMSE)) are corrected after cross-validation of the model. The results of processing by comparing with various machine learning algorithms, show that the greatest value in application 1 is obtained for the Linear Regression model compared to other models, meanwhile application 2 and 3 show that the greatest value is obtained for the kNN model compared to other models.

Model development can be improved by making a performance log that is adjusted to the performance of the resulting model. This means that the log feature can be added or subtracted according to the model produced, to get a better model in the future.

## References

1. Huang CJ, Wang YW, Guan CT, Chen HM, Jian JJ. Applications of machine learning to resource management in cloud computing. International Journal of Modeling and Optimization. 2013; 3(2): p. 148.
2. Berral JL, Goiri I, Nou R, Julià F. Toward energy-aware scheduling using machine learning. Energy Efficient Distributed Computing Systems. 2012;: p. 215-244.
3. Roy C, Moitra S, Das M, Srinivasan S, Malhotra R. IT Infrastructure Downtime Preemption using Hybrid Machine Learning and NLP. In

FedCSIS (Position Papers). 2015: p. 39-44.

4. Gao J. Machine learning applications for data center optimization. 2014.
5. Fang Z. Resource management on cloud systems with machine learning. MS thesis. Universitat Politècnica de Catalunya; 2010.
6. Alonso J, Torres J, Gavalda R. Predicting web server crashes: A case study in comparing prediction algorithms. In In 2009 Fifth International Conference on Autonomic and Autonomous Systems; 2009: IEEE. p. 264-269.
7. Zhang Q, Cherkasova L, Mi N, Smirni E. A regression-based analytic model for capacity planning of multi-tier applications. Cluster Computing. 2008; 11(3): p. 197-211.
8. Kapadia NH, Fortes JA, Brodley CE. Predictive application-performance modeling in a computational grid environment. In The Eighth International Symposium on High Performance Distributed Computing; 1999: IEEE. p. 47-54.
9. Verma N, Sharma A. Workload prediction model based on supervised learning for energy efficiency in cloud. In In 2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA) ; 2017. p. 66-71.
10. Tseng FH, Wang X, Chou LD, Chao HC, Leung VC. Dynamic resource prediction and allocation for cloud data center using the multiobjective genetic algorithm. IEEE Systems Journal. 2017; 12(2): p. 1688-1699.
11. Shyam GK, Manvi SS. Virtual resource prediction in cloud environment: a Bayesian approach. Journal of Network and Computer Applications. 2016;: p. 144-154.
12. Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction: Springer Science & Business Media; 2009.
13. Bowles M. Machine learning in Python: essential techniques for predictive analysis Indianapolis: IN: John Wiley & Sons, Inc.; 2015.
14. James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning New York: Springer; 2013.
15. Shen Y. Virtual resource scheduling prediction based on a support vector machine in cloud computing. In In 2015 8th International Symposium on Computational Intelligence and Design (ISCID); 2015. p. 110-113.