Performance Comparison between Naïve Bayes, Decision Tree and k-Nearest Neighbor in Searching Alternative Design in an Energy Simulation Tool

Ahmad Ashari
Department of Computer Science and Electronics
GadjahMada University
Yogyakarta, Indonesia

Iman Paryudi
Institute of Software Technology and Interactive Systems
Vienna University of Technology
Vienna, Austria

A Min Tjoa
Institute of Software Technology and Interactive Systems
Vienna University of Technology
Vienna, Austria

***Abstract*—Energy simulation tool is a tool to simulate energy use by a building prior to the erection of the building. Commonly it has a feature providing alternative designs that are better than the user's design. In this paper, we propose a novel method in searching alternative design that is by using classification method. The classifiers we use are Naïve Bayes, Decision Tree, and k-Nearest Neighbor.**

Our experiments hows that Decision Tree has the fastest classification time followed by Naïve Bayes and k-Nearest Neighbor. The differences between classification time of Decision Tree and Naïve Bayes also between Naïve Bayes and k-NN are about an order of magnitude. Based on Percision, Recall, F-measure, Accuracy, and AUC, the performance of Naïve Bayes is the best. It outperforms Decision Tree and k-Nearest Neighbor on all parameters but precision.

Keywords—energy simulation tool; classification method; naïve bayes; decision tree; k-nearest neighbor

I. Introduction

Energy simulation tool is a tool to simulate energy use by a building prior to the erection of the building. The output of such simulation is a value in kWh/m^2 called energy performance. The calculation of the building energy performance must be carried out by developers as part of requirements to get permit to build the building. The building can only be built if the energy performance is below the allowable standard.

In order to get building energy performance below the standard, architects must revise the design several times. And in order to ease the design work of the architects, an energy simulation tool must have a feature that suggests a better alternative design.

Since the alternative design search is actually a classification problem, hence in this paper we propose a novel method to search alternative design by using classification method. The classification methods used in here are Decision Tree, Naïve Bayes, and k-Nearest Neighbor. We will then compare the performance of these three methods in searching alternative design in an energy simulation tools.

The rest of the paper is structured as follows: Section 2 describes the classification methods we use in this study. Section 3 explains the data preparation followed by the experiment in Section 4. The result and its discussion are presented in section 5 and 6 respectively. Section 7 concludes the paper.

II. Classification Method

Classification is the separation or ordering of objects into classes [1]. There are two phases in classification algorithm: first, the algorithm tries to find a model for the class attribute as a function of other variables of the datasets. Next, it applies previously designed model on the new and unseen datasets for determining the related class of each record [2].

Classification has been applied in many fields such as medical, astronomy, commerce, biology, media, etc. There are many techniques in classification method like: Decision Tree, Naïve Bayes, k-Nearest Neighbor, Neural Networks, Support Vector Machine, and Genetic Algorithm. In this paper we will use Decision Tree, Naïve Bayes, and k-Nearest Neighbor.

A. Decision Tree

A decision tree is a flow-chart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and leaf nodes represent classes or class distributions [3].

The popular Decision Tree algorithms are ID3, C4.5, CART. The ID3 algorithm is considered as a very simple decision tree algorithm. It uses information gain as splitting criteria. C4.5 is an evolution of ID3. It uses gain ratio as splitting criteria [4].

CART algorithm uses Gini coefficient as the test attribute selection criteria, and each time selects an attribute with the smallest Gini coefficient as the test attribute for a given set [5].

The advantage of using Decision Trees in classifying the data is that they are simple to understand and interpret [6]. However, decision trees have such disadvantages as [4]:

1) Most of the algorithms (like ID3 and C4.5) require that the target attribute will have only discrete values.
2) As decision trees use the "divide and conquer" method, they tend to perform well if a few highly relevant attributes exist, but less so if many complex interactions are present.

B. Naive Bayes

Naïve Bayesian classifiers assume that there are no dependencies amongst attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, hence is called "naive" [3]. This classifier is also called idiot Bayes, simple Bayes, or independent Bayes [7].

The advantages of Naive Bayes are [8]:

- It uses a very intuitive technique. Bayes classifiers, unlike neural networks, do not have several free parameters that must be set. This greatly simplifies the design process.
- Since the classifier returns probabilities, it is simpler to apply these results to a wide variety of tasks than if an arbitrary scale was used.
- It does not require large amounts of data before learning can begin.
- Naive Bayes classifiers are computationally fast when making decisions.

C. k-Nearest Neighbor

The *k*-nearest neighbor algorithm (k-NN) is a method to classify an object based on the majority class amongst its *k*-nearest neighbors. The k-NN is a type of lazy learning where the function is only approximated locally and all computation is deferred until classification [9].

k-NN algorithm usually use the Euclidean or the Manhattan distance. However, any other distance such as the Chebyshev norm or the Mahalanob is distance can also be used [10]. In this experiment, Euclidean distance is used. Suppose the query instance have coordinates (a, b) and the coordinate of training sample is (c, d) then square Euclidean distance is:

$$x^2 = (c - a)^2 + (d - b)^2 \qquad (1)$$

III. Data Preparation

In classification method, training set is needed to construct a model. This training set contains a set of attributes with one attribute being the attribute of the class. Then the constructed model is used to classify an instance.

For this experiment, there are more than 67 millions of raw data available. This data comes from combination of 13 building parameters with each parameter has 4 possible values (4^{13} data).

The parameters and the values used in each parameter are as follows:

1. Wall U-value: 0.1; 0.15; 0.2; 0.25 W/m^{2}K
2. Wall Height: 2.5; 3.0; 3.5; 4.0 m
3. Roof U-value: 0.1; 0.15; 0.2; 0.25 W/m^{2}K
4. Floor U-value: 0.1; 0.15; 0.2; 0.25 W/m^{2}K
5. Floor Area: 70; 105; 140; 175 m^2
6. Number of Floors: 1; 2; 3; 4
7. Window U-value: 0.1; 0.7; 1.3; 1.9 W/m^{2}K
8. South Window Area: 0; 4; 8; 12 m^2
9. North Window Area: 0; 4; 8; 12 m^2
10. East Window Area: 0; 4; 8; 12 m^2
11. West Window Area: 0; 4; 8; 12 m^2
12. Door U-value: 0.1; 0.7; 1.3; 1.9 W/m^{2}K
13. Door Area: 2; 4; 6; 8 m^2

Since the data is very big, representative training set must be selected. Besides that the training set must be as small as possible. With the above considerations in mind, 5 candidate training sets created. They are with different number of data. The candidate training sets are:

- Training set 1: 2827 data
- Training set 2: 4340 data
- Training set 3: 5405 data
- Training set 4: 6819 data
- Training set 5: 8630 data

To select the best training set, an experiment using the three classifiers is carried out. The experiment is done by means of Weka data mining software. For this experiment we use 10-fold cross validation. The results are depicted in Fig. 1, 2, and 3.

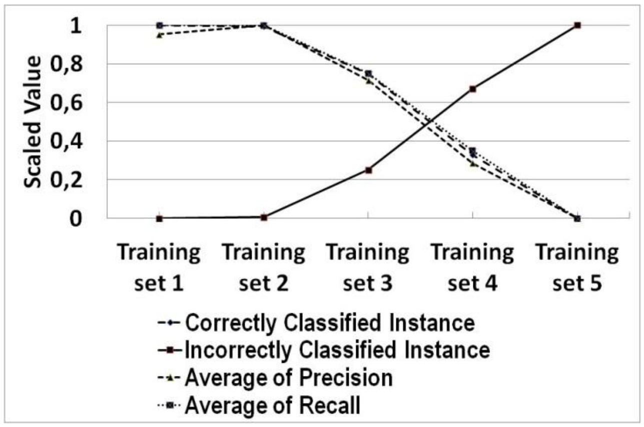

Fig. 1. k-NN performance on different training sets.

Fig. 1 shows performance of k-NN methods using the five training sets. The classifier shows the best performance when using training sets 1 and 2. However, k-NN performance has better precision when using training set 2 than training set 1. Fig. 2 shows performance of Naïve Bayes classifier using the same training sets. Naïve Bayes performs best when using training set 2. This is shown by the highest correctly classified instance and precision, and the lowest incorrectly classified instance.

Meanwhile Fig. 3 shows no performance difference on Decision Tree when using the training sets. From this result, training set 2 is chosen as the working training set.

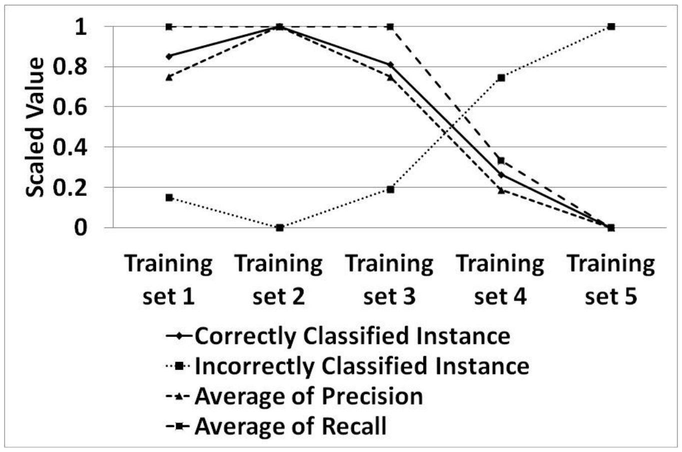

Fig. 2. Naïve Bayes performance on different training sets.

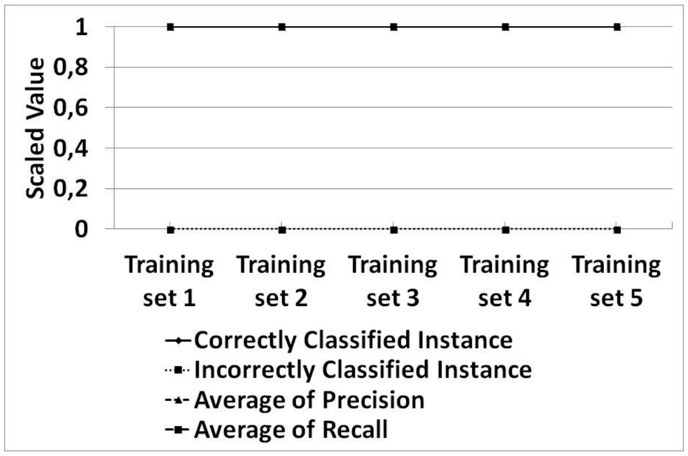

Fig. 3. Decision Tree performance on different training sets.

IV. Experiment

To carry out the experiment, a simple energy simulation tool using the three classifiers (Naïve Bayes, Decision Tree, and k-NN) is developed. For the Decision Tree we use C4.5 algorithm and for k-NNwe use k = 11.We did an experiment using 10 data and for each data, a classification time and performance values are recorded. We should mention here that the time we use is classification time only (without training time). The reason is that K-NN is lazy learner that does not need training. Hence to be fair, the time we use here is only classification time.

Except classification time, the output of the experiment is a confusion matrix. Using confusion matrix, performance parameters of a classifier can be calculated. The performance parameters include: precision, recall, accuracy, F-measure, and area under the curve (AUC).

We use AUC in this experiment because Provost et al., 1998 in [11] state that simply using accuracy results can be misleading. They recommended when evaluating binary decision problems to use Receiver Operator Characteristic (ROC) curves, which show how the number of correctly classified positive examples varies with the number of incorrectly classified negative examples. This is supported byEntezari-Maleki, Rezaei, Minaei-Bidgoli [12]who state that ROC curve is a usual criterion for identifying the prediction power of different classification methods, and the area under this curve is one of the important evaluation metrics which can be applied for selecting the best classification method.

An ROC graph isactually two-dimensional graph in which True Positive Rate (TPR) is plotted on the Y axis and False Positive Rate (FPR) is plotted on the X axis [13]. It depicts relative trade-offs between benefits (true positives) and costs (false positives). One point in ROC space is better than another if its TPR is higher,FPR is lower, or both[14]. ROC performance of a classifier is usually represented by a value which is the area under the ROC curve (AUC). The value of AUC is between 0 and 1.

The experiment steps are as follows:

1) Enter user data. Values of all 13 parameters are entered. The application then calculates the energy performance. For instance the energy performance of the user data is X W/m^2. The energy performance is calculated using the following formulas:

$$Le = 1.0 * (wa - wina - da) * wuv + 1.0 * wina * winuv + 1.0 * da * duv \quad (2)$$

$$Lu = 0.9 * ra * ruv \quad (3)$$

$$Lg = 0.5 * fa * fuv \quad (4)$$

$$tl = Le + Lu + Lg \quad (5)$$

$$TL = 0.024 * tl * 3235 \quad (6)$$

$$Lv = 0.33 * 0.6 * fa * wh * 0.8 \quad (7)$$

$$VL = 0.024 * Lv * 3235 \quad (8)$$

$$IG = 0.024 * 4 * fa * nof * 208 \quad (9)$$

$$SG = 356 * (swa * 0.75) * 0.9 * 0.67 * 0.9 + 150 * (nwa * 0.75) * 0.9 * 0.67 * 0.9 + 210 * (ewa * 0.75) * 0.9 * 0.67 * 0.9 + 210 * (wwa * 0.75) * 0.9 * 0.67 * 0.9 \quad (10)$$

$$EP = (TL + VL) - 1.0 * (IG + SG) \quad (11)$$

where:

Le = exterior loss

wa = wall area

wina = window area

da = door area

wuv = wall u-value

winuv = window u-value

duv = door u-value

Lu = unheated space loss

ra = roof area

ruv = roof u-value

Lg = ground loss

fa = floor area

fuv = floor u-value

tl = thermal loss

TL = transmission loss

wh = wall height

VL = ventilation loss

IG = internal gain

nof = number of floors

SG = solar gain

swa = south window area

nwa = north window area

ewa = east window area

wwa = west window area

EP = energy performance

2) Setting classes of the training set. Every data in the training set having energy performance less than or equal to X W/m^2 is set to class Good, and those having energy performance greater than X W/m^2 is set to class Bad.Note that the attributes of training set are: Wall U-value, Wall Height, Roof U-value, Floor U-value, Floor Area, Number of Floors, Window U-value, South Window Area, North Window Area, East Window Area, West Window Area, Door U-value, Door Area, Energy Performance, Class.

3) Create working data.The working data is created by querying on the raw data. Since there are 13 parameters, there will be 13 queries. The condition on each query is taken from the value of the respective parameter on the user data.The queries are done one after another. It means that the data resulted from a query will be queried again by the next query.This is done 13 times.Note that the attributes of working data are:Wall U-value, Wall Height, Roof U-value, Floor U-value, Floor Area, Number of Floors, Window U-value, South Window Area, North Window Area, East Window Area, West Window Area, Door U-value.

4) Classification. Data from working data is taken one by one. This data is then classified against the training set using one of the three classifiers (Naïve Bayes, Decision Tree, k-Nearest Neighbor). The classification time is recorded starting from the beginning until the end of the classification. After the classification, the energy performance of this data is calculated. Note that the data resulted in this step has the following attributes: Wall U-value, Wall Height, Roof U-value, Floor U-value, Floor Area, Number of Floors, Window U-value, South Window Area, North Window Area, East Window Area, West Window Area, Door U-value, Door Area, Energy Performance, Class, Classification time.

5) Create confusion matrix. Count True Positive (TP), False Positive (FP), True Negative (TN), False Negative (FN). A data is included in TP if it has energy performance less than or equal to X W/m^2 and class Good. A data is included in TN if it has energy performance greater than X W/m^2 and class Bad. A data is included in FP if it has energy performance greater than X W/m^2 but has class Good. Meanwhile a data is included in FN if it has energy performance less than or equal to X W/m^2 but has class Bad.

6) Select alternative design. Of all data included in TP, the one having the best energy performance will be selected as the alternative design.

V. RESULT

The classification times of the three classifiers that are used to classify 10 data are shown in Fig. 4.This figure shows that Decision Tree has the fastest classification time followed by Naïve Bayes and k-Nearest Neighbor. The differences between classification time of Decision Tree and Naïve Bayes also between Naïve Bayes and k-NN are about an order of magnitude.

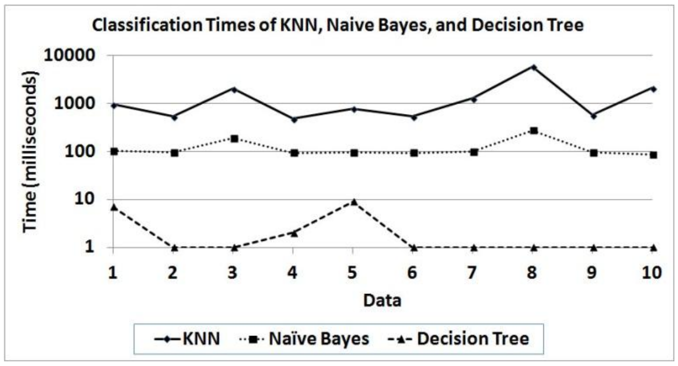

Fig. 4. Classification times of k-NN, Naïve Bayes, and Decision Tree.

The average precisions and recalls for k-NN, Naïve Bayes, and Decision Tree are: 0.819 and 0.543; 0.799 and 0.794; 0.779 and 0.663 respectively(Fig. 5 and 6). Since F-measure is the harmonic mean of precision and recall, hence to know which classifier is the best in terms of precision and recall, we can calculate the F-measure value (Fig. 7). The average F-measure value of Naïve Bayes is the biggest among the three, that is 0.780. Decision tree has average F-measure of 0.676 and k-NN of 0.543. Therefore we can say that Naïve Bayes is the best in terms of precision and recall followed by Decision Tree and k-NN.

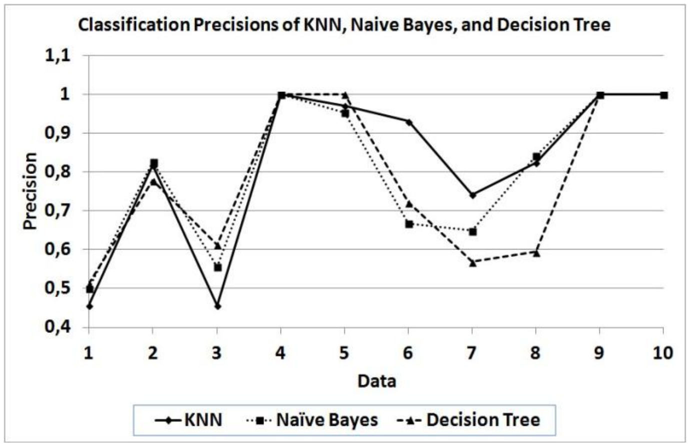

Fig. 5. Classification precision of k-NN, Naïve Bayes, and Decision Tree

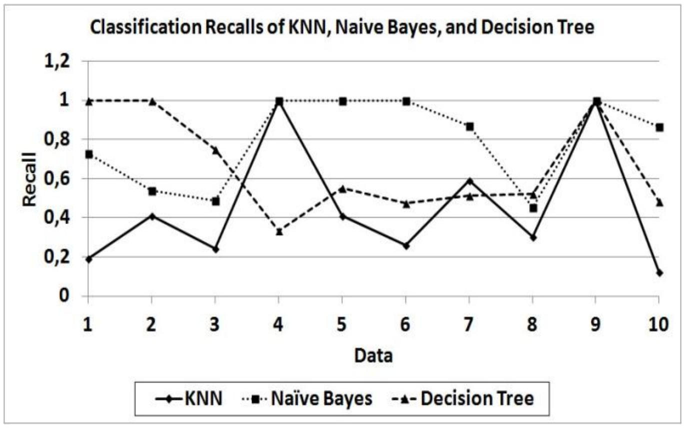

Fig. 6. Classification recall of k-NN, Naïve Bayes, and Decision Tree

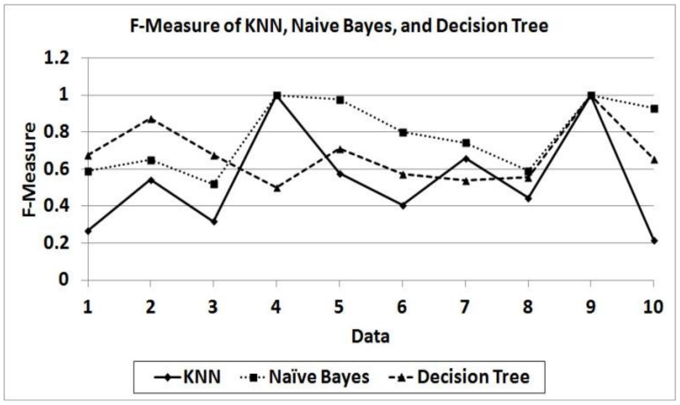

Fig. 7. F-measure of k-NN, Naïve Bayes, and Decision Tree

Naïve Bayes is again the best in accuracy (Fig. 8). Naïve Bayes is the most accurate classifier compared to Decision Tree and k-NN with the average accuracy of 0.737. Meanwhile the average accuracies of Decision Tree and k-NN are 0.589 and 0.567, respectively.

The last parameter for comparing classifier performance is area under the curve (AUC). In this parameter Naïve Bayes is also the biggest among the three classifiers (Fig. 9). The AUC of Naïve Bayes is 0.605, followed by Decision Tree 0.585 and k-NN 0.570.

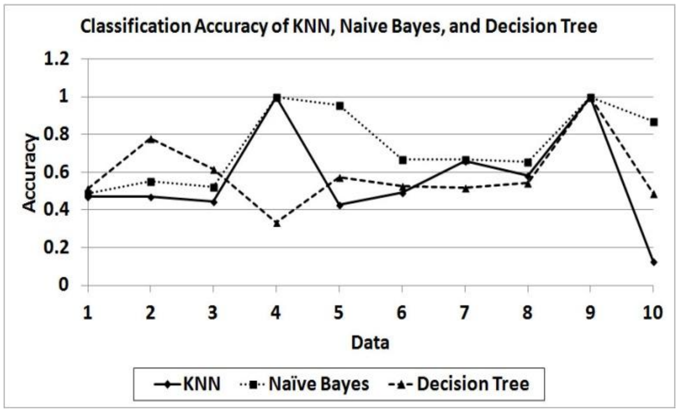

Fig. 8. Classification accuracy of k-NN, Naïve Bayes, and Decision Tree

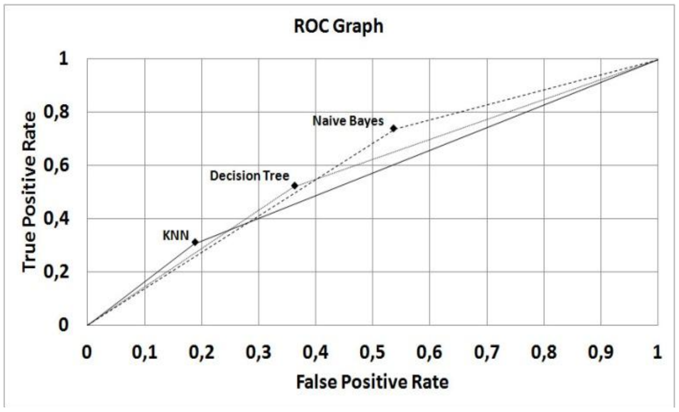

Fig. 9. Area under the curve (AUC) of k-NN, Naïve Bayes, and Decision Tree

VI. Discussion

As stated in the previous section, the experiment we carried out reveals that Naïve Bayes outperforms Decision Tree and k-NN. It is the best in all performance parameters but precision, they are: recall, F-measure, accuracy, and AUC. This result is similar to previous studies.

When comparing Naïve Bayes and Decision Tree in the classification of training web pages, Xhemali,Hinde, and Stone[15] find that the accuracy, F-measure, and AUC of Naïve Bayes are 95.2, 97.26, and 0.95 respectively. This is better than Decision Tree whose accuracy, F-measure, and AUC are: 94.85, 95.9, 0.91, respectively.

Li and Jain [16] investigate four different methods for document classification: the naive Bayes classifier, the nearest neighbour classifier, decision trees and a subspace method. Their experimental results indicate that the naive Bayes classifier and the subspace method outperform the other two classifiers on the data sets. Their experimental results show that all four classification algorithms perform reasonably well; the naïve Bayes approach performs the best on test data set1, but the subspace method outperforms all others on test data set2.

Other studies in references [17] - [20] also obtain the same results when comparing performance of Naïve Bayes and Decision Tree.

A Naive Bayes classifier is a simple classifier. However, although it is simple, Naive Bayes can outperform more sophisticated classification methods. Besides that it has also exhibited high accuracy and speed when applied to large database [3]. Moreover, it is very fast for both learning and predicting. Its learning time is linear in the number of examples and its prediction time is independent of the number of examples [21].Naïve Bayes classifier is also fast, consistent, easy to maintain and accurate in the classification of attribute data [15]. And from computation point of view, Naïve Bayes is more efficient both in the learning and in the classification task than Decision Tree [22].

The reason for good performance of Naïve Bayes is described by Dominggos and Pazzani [23]as follows:"Naïve Bayes is commonly thought to be optimal, in the sense of achieving the best possible accuracy, only when the independence assumption holds, and perhaps close to optimal when the attributes are only slightly dependent. However, this very restrictive condition seems to be inconsistent with the Naïve Bayes' surprisingly good performance in a wide variety of domains, including many where there are clear dependencies between the attributes." In a study on 28 datasets from the UCI repository, they find that Naïve Bayes was more accurate than C4.5 in 16 domains. They further statethat: "the Naïve Bayes is in fact optimal even when the independence assumption is grossly violated, and is thus applicable to a much broader range of domains than previously thought. This is essentially due to the fact that in many cases the probability estimates may be poor, but the correct class will still have the highest estimate, leading to correct classification". Finally they come to conclusion that "the Naïve Bayes achieves higher accuracy than more sophisticated approaches in many domains where there is substantial attribute dependence, and therefore the reason for its good comparative performance is not that there are no attribute dependences in the data".

Frank, Trigg, Holmes, and Witten[24] explain why naive Bayes perform well even when the independence assumption is seriously violated: "most likely it owes its good performance to the zero-one loss function used in classification. This function defines the error as the number of incorrect predictions. Unlike other loss functions, such as the squared error, it has the key property that it does not penalize inaccurate probability estimates as long as the greatest probability is assigned to the correct class. There is evidence that this is why naive Bayes' classification performance remains high, despite the fact that inter-attribute dependencies often cause it to produce incorrect probability estimates".

Meanwhile Zhang [25] explains the reason of good performance of Naïve Bayes as follows:"In a given dataset, two attributes may depend on each other, but the dependence may distribute evenly in each class. Clearly, in this case, the conditional independence assumption is violated, but naive Bayes is still the optimal classifier. Further, what eventually affects the classification is the combination of dependencies among all attributes. If we just look at two attributes, there may exist strong dependence between them that affects the classification. When the dependencies among all attributes work together, however, they may cancel each other out and no longer affect the classification". Therefore, he argues that "it is the distribution of dependencies among all attributes over classes that affect the classification of naive Bayes, not merely the dependencies themselves".

Similar to the result of our study, previous studies also show that k-Nearest Neighbor is worse than both Naïve Bayes and Decision Tree. In their study to classify arid rangeland using Decision Tree and k-Nearest Neighbor, Laliberte, Koppa, Fredrickson, and Rango[26] obtain that the overall accuracy of Decision Tree (80%) is better than that of k-Nearest Neighbor (78%). Pazzani,Muramatsu, and Billsus[27] find that in identifying interesting web sites, the naive Bayesian classifier has the highest average accuracy with 20 training examples: 77.1 (standard deviation 4.4). In contrast, backprop is 75.0 (3.9), k-Nearest Neighbor is 75.0 (5.5), and ID3 is 70.6 (3.6). The only study which shows that k-NN outperforms Decision Tree and Naïve Bayes is by Horton and Nakai[28]. However, they do not have a solid answer as to why k-NN performs better on this task.

The performance of k-NN in this and previous studies is the worst among the three classifiers. Since k-NN uses number of nearest neighbor k as one of the parameter in classifying an object, then this value might affect the performance of the classifier. In their study using k-NN to classify credit card applicants, Islam,Wu, Ahmadi, Sid-Ahmed[29] find that the best performance of k-NN is when k=5. Using this k value, k-NN outperforms Naïve Bayes. Using bigger and smaller k value, the k-NN performance is worst. Meanwhile, Batista and Silva [30] study three parameters affecting the performance of k-NN, namely number of nearest neighbors (k), distance function, and weighting function. They find that for all weighting function and distance function, the performance increases as k increases up to a maximum between k = 5 and k = 11. Then, for higher values of k, the performance decreases. Based on this study, we use k = 11 in this experiment. And the reason why we choose the upper boundary is because larger k values help reduce the effects of noisy points within the training data set [29].The choice is also based on our experiment onk-NN performance with different k values. The k values we use are: 11, 21, 31, 41, and 51. The experiment use 10-fold cross validation. The result is shown in Fig. 10. The figure shows thatk-NN reaches the best performance when we use k = 11. For k values greater than 11, the performance decreases. Since we have not tested the k values smaller than 11, hence it is worth trying to use those values in the future work.

Beside low performance, another weakness of k-NN is slow runtime performance and large memory requirements [31]. The k-NN classifier requires a large memory to store the entire training set [32]. Hence, the bigger the training set, the bigger memory requirement and the larger distance calculations must be performed. This causes the classification is extremely slow. This is the reason why the classification time of k-NN in our experiment is very big, the worst among the three classifiers.

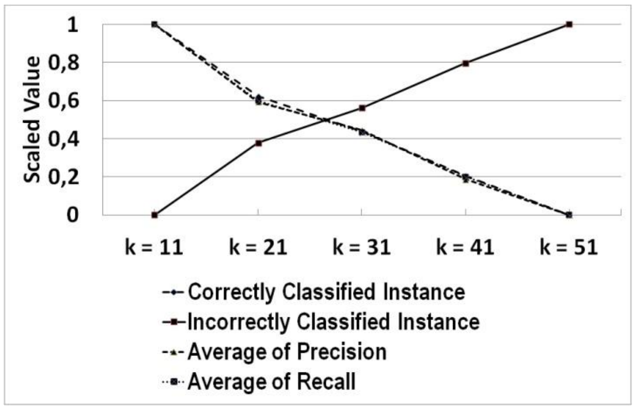

Fig. 10. k-NN performance on different k values.

The fast classification time by Decision Tree is due to the absence of calculation in its classification process. The tree model is created outside the application, using Weka data mining tool. And the model is converted into rules before being incorporated into the application. Classification by way of following the tree rules is faster than the ones that need calculation as in the case of Naïve Bayes and k-NN.

VII. CONCLUSION

A novel method to search alternative design in an energy simulation tool is proposed. A classification method is used in searching the alternative design. There are three classifiers used in this experiment namely Naïve Bayes, Decision Tree, and k-Nearest Neighbor. Our experiment shows that Decision Tree is the fastest and k-Nearest Neighbor is the slowest. The fast classification time of Decision Tree because there is no calculation in its classification. The tree model is created outside the application that is using Weka data mining tool. And the model is converted into rules before being incorporated into the application. Classification by way of following the tree rules is faster than the ones that need calculation as in the case of Naïve Bayes and k-NN. Meanwhile k-Nearest Neighbor is the slowest classifier because the classification time is directly related to the number of data. The bigger the data, the larger distance calculations must be performed. This causes the classification is extremely slow.

Although it is a simple method, Naïve Bayes can outperform more sophisticated classification methods. In this experiment, Naïve Bayes outperforms Decision Tree and k-Nearest Neighbor. Dominggos and Pazzani[23] state that the reason for Naïve Bayes' good performance is not because there are no attribute dependences in the data. In fact Frank,Trigg, Holmes, and Witten[24] explain that its good performance is caused by the zero-one loss function used in the classification. Meanwhile Zhang [25] argues that it is the distribution of dependencies among all attributes over classes that affect the classification of naive Bayes, not merely the dependencies themselves.

ACKNOWLEDGMENT

ImanParyudi would like to thank Directorate of Higher Education, Ministry of Education and Culture, Republic of Indonesia for the scholarship awarded to him.

REFERENCES

[1] G. K. Gupta, Introduction to Data Mining with Case Studies. Prentice Hall of India, New Delhi, 2006.

[2] P-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining. Addison Wesley Publishing, 2006.

[3] J. Han and M. Kamber, Data Mining: Concepts and Techniques. Morgan-Kaufmann Publishers, San Francisco, 2001.

[4] O. Maimon and L. Rokach, Data Mining and Knowledge Discovery. Springer Science and Business Media, 2005.

[5] X. Niuniu and L. Yuxun, "Review of Decision Trees," IEEE, 2010.

[6] V. Mohan, "Decision Trees: A comparison of various algorithms for building Decision Trees," Available at: http://cs.jhu.edu/~vmohan3/document/ai_dt.pdf

[7] T. Miquelez, E. Bengoetxea, P. Larranaga, "Evolutionary Computation based on Bayesian Classifier," Int. J. Appl. Math. Comput. Sci. vol. 14(3), pp. 335 – 349, 2004.

[8] M. K. Stern, J. E. Beck, and B. P. Woolf, "Naïve Bayes Classifiers for User Modeling," Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.118.979

[9] Wikipedia, "k-Nearest Neighbor Algorithm," Available at: http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm

[10] V. Garcia, C. Debreuve, "Fast k Nearest Neighbor Search using GPU," IEEE, 2008.

[11] J. Davis and M. Goadrich, "The Relationship Between Precision-Recall and ROC Curves," Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, 2006.

[12] R. Entezari-Maleki, A. Rezaei, and B. Minaei-Bidgoli, "Comparison of Classification Methods Based on the Type of Attributes and Sample Size," Available at: http://www4.ncsu.edu/~arezaei2/paper/JCIT4-184028_Camera%20Ready.pdf

[13] Wikipedia, "Receiver Operating Characteristics," Available at: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

[14] T. Fawcett, "ROC Graphs: Notes and Practical Considerations for Researchers," Kluwer Academic Publishers, Netherland, 2004.

[15] D. Xhemali, C. J. Hinde, and R. G. Stone, "Naïve Bayes vs. Decision Trees vs. Neural Networks in the Classification of Training Web Pages," International Journal of Computer Science Issue, Vol. 4(1), 2009.

[16] Y. H. Li and A. K. Jain, "Classification of Text Document," The Computer Journal, Vol. 41(8), 1998.

[17] R. M. Rahman and F. Afroz, "Comparison of Various Classification Techniques," Journal of Software Engineering and Applications, Vol. 6, 2013, 85 – 97.

[18] Z. Nematzadeh Balagatabi, "Comparison of Decision Tree and Naïve Bayes Methods in Classification of Researcher's Cognitive Styles in Academic Environment," Journal of Advances in Computer Research. Vol. 3(2), 2012, 23 – 34.

[19] L. Dan, L. Lihua, Z. Zhaoxin, "Research of Text Categorization on WEKA," Third International Conference on Intelligent System Design and Engineering Applications, 2013.

[20] J. Huang, J. Lu, C. X. Ling, "Comparing Naive Bayes, Decision Trees, and SVM with AUC and Accuracy," Third IEEE International Conference on Data Mining, 2003.

[21] M. Pazzani and D. Bilsus, "Learning and Revising User Profiles: The Identification of InterestingWeb Sites," Machine Learning, Vol. 27, 313 – 331, 1997.

[22] N. B. Amor, S. Benferhat, Z. Elouedi, "Naive Bayes vs Decision Trees in Intrusion Detection Systems," ACM, 2004.

[23] P. Domingos, M. Pazzani, "Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier," Available at: http://www.ics.uci.edu/~pazzani/Publications/mlc96-pedro.pdf

[24] E. Frank, L. Trigg, G. Holmes, I. A. Witten, "Naïve Bayes for Regression," Machine Learning, Vol. 000, 1 – 20, 1999.

[25] H. Zhang, "The Optimality of Naïve Bayes," American Association for Artificial Intelligence, 2004.

[26] A. S. Laliberte, J. Koppa, E. L. Fredrickson, and A. Rango, "Comparison of nearest neighbor and rule-based decision tree classification in an object-oriented environment," Available at: http://naldc.nal.usda.gov/download/44074/PDF

[27] M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & Webert: Identifying interesting web sites," Available at: http://www.ics.uci.edu/~pazzani/RTF/AAAI.html

[28] P. Horton and K. Nakai, "Better Prediction of Protein Cellular Localization Sites with the k Nearest Neighbors Classifier," ISMB-97 Proceedings, AAAI, 1997.

[29] M. J. Islam, Q. M. J. Wu, M. Ahmadi, M. A. Sid-Ahmed, "Investigating the Performance of Naïve- Bayes Classifiers and K- Nearest Neighbor Classifiers," Journal of Convergence Information Technology, Vol. 5(2), 2010.

[30] G. E.A.P.A. Batista, D. F. Silva, "How k-Nearest Neighbor Parameters Affect its Performance," Simposio Argentino de Inteligencia Artificial (ASAI 2009), 95 – 106, 2009.

[31] S. D. Bay, "Nearest Neighbor Classification from Multiple Feature Subsets," Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.51.9040&rep=rep1&type=pdf

[32] Y. Lee, "Handwritten Digit Recognition Using K Nearest-Neighbor, Radial-Basis Function, and Backpropagation Neural Networks," Neural Computation Vol. 3, 440 – 449, 1991