International Journal of Engineering & Technology, 8 (1.7) (2019) 214-218

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

E-Commerce Product Classification Using Supervised Learning Models

Norsyela Muhammad Noor Mathivanan [1] , Nor Azura Md. Ghani [1,2*], Roziah Mohd Janor [3]

[1]Center for Statistical and Decision Sciences Studies, Faculty of Computer & Mathematical Sciences, 40450 Shah Alam, Universiti Teknologi MARA, Malaysia
[2]National Design Centre, Universiti Teknologi MARA, 40450 Shah Alam, Selangor, MALAYSIA
**Corresponding author E-mail: azura@tmsk.uitm.edu.my*

Abstract

E-commerce has become a major player in today's marketplace having a large database of products and number of retailers and consumers use these services. However, these products are placed into different categories according to the structure of different websites. An automatic classification model helps in classifying the products efficiently. This paper presents a comparative study on different algorithms from supervised learning model to classify real-world datasets related to e-commerce products. The results show that KNN is the best model with the highest accuracy to classify the data used in the study. Hence, KNN model is a good approach in classifying e-commerce products.

Keywords: *Text Classification, E-commerce product, Supervised Learning Model*

1. Introduction

Nowadays, most retailers prefer to use the exciting environment of electronic commerce also known as e-commerce platforms in promoting their products and services. It encloses a wide range of interaction processes between various market users include order, delivery, invoice and payment processes [1]. The growth of e-commerce is contributed the most by the easiness of transacting money over internet and the availability of various kind of products [2]. There are large inventories with millions of products sold in the online marketplaces. These products are posted across different countries through e-commerce websites such as 11street, Amazon, e-Bay and Alipay. Consumers are able to view many new products in these websites over time. Most of the websites are well-structured and they consist of product information such as the product name, description, price, and image. However, most of the products are assigned into desired categories manually by the retailers. Thus, the availability of classification models may help in categorizing the products automatically.

The classification of e-commerce products can be done using supervised learning model. A supervised learning model is fairly common in solving classification problems because the goal is to acquire the computer to learn a classification system that has been created. Various types of supervised learning models have been used in many fields of studies such as market segmentation [3], natural language processing [4], bioinformatics [5] and pattern recognition [6]. However, the comparison between well-known supervised learning models including Naïve Bayes, K-Nearest Neighbor (KNN), Decision Trees, Support Vector Machine (SVM) and Random Forest is not yet to be seen in a research related to product classification. It is important to assess the performance of each model because the results provide valuable information regarding the best model to classify this kind of data.

Hence, this paper aims to compare the performance of supervised learning models in classifying different categories of e-commerce products. The rest of this paper is organized as follows: Section 2 describes the supervised learning models used in the research, the dataset used and research design; Section 3 presents the performance measures and the corresponding evaluation results; Section 4 concludes the research and directions for future work.

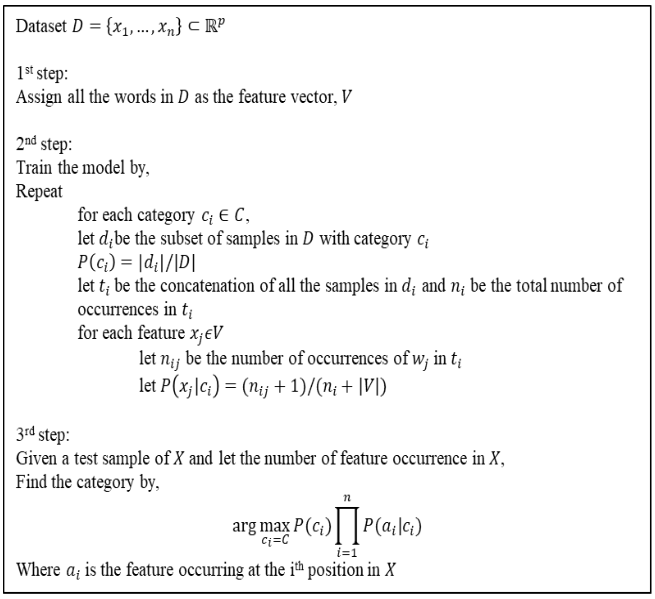
2. Method

2.1. Supervised Learning Model

Supervised learning model is used to make predictions based on information about the targets and the features of data. It infers a function according to a given set of input-output data respectively. Normally, the input data provides a set of observations with which the computer is trained [7]. Each observation consists of an input vector and a desired output value. A supervised learning model trains the data and generates a general rule or function to be used for predicting or classifying new inputs.

2.1.1. Naïve Bayes

Naïve Bayes is a classification model based on Bayes theorem introduced by Thomas Bayes and it has been used as conventional paradigm since late 18th century [8]. It is one of probabilistic-based classifiers where it predicts the probability of the sample itself before choosing the class with highest probability given the observation. It is widely used in text categorization, sentiment analysis and spam filtering [9]. The algorithm for Naïve Bayes [10] is given in Fig. 1.

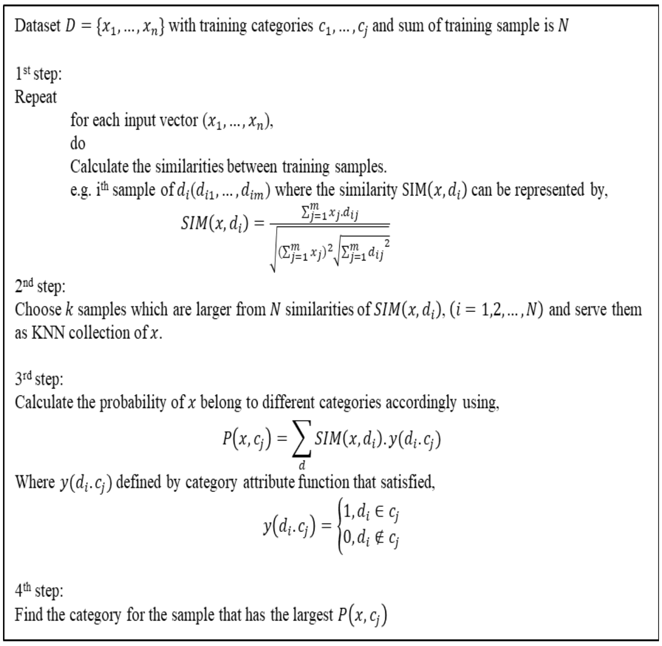


```
Dataset $D = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$

1st step:
Assign all the words in $D$ as the feature vector, $V$

2nd step:
Train the model by,
Repeat
        for each category $c_i \in C$,
        let $d_i$ be the subset of samples in $D$ with category $c_i$
        $P(c_i) = |d_i|/|D|$
        let $t_i$ be the concatenation of all the samples in $d_i$ and $n_i$ be the total number of
        occurrences in $t_i$
        for each feature $x_j \epsilon V$
                let $n_{ij}$ be the number of occurrences of $w_j$ in $t_i$
                let $P(x_j|c_i) = (n_{ij} + 1)/(n_i + |V|)$

3rd step:
Given a test sample of $X$ and let the number of feature occurrence in $X$,
Find the category by,
                $\arg\max_{c_i=C} P(c_i) \prod_{i=1}^{n} P(a_i|c_i)$
Where $a_i$ is the feature occurring at the ith position in $X$
```

Fig. 1: Naïve Bayes Algorithm

2.1.2. K-Nearest Neighbor

K-Nearest Neighbor (KNN) is the fundamental classification model to classify observations according to closest training examples in the feature space when there is little or no prior knowledge on the distribution of the data [11]. It is an instance-based learning where the function is close to local value and the computations are deferred before the classifying process occurs. Basically, the rule holds the training set as a whole during the learning process. Then, each observation is assigned to a class according to the majority label of its KNN in the training dataset. A sample should be grouped into its similar surrounding samples. Thus, the nearest neighbor samples can be considered to classify or predict an unknown sample. The algorithm for KNN [12] is given in Fig. 2.

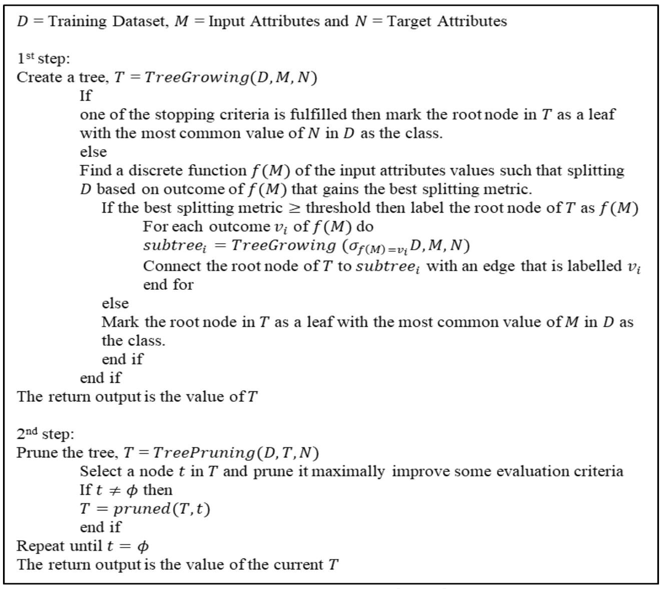


```
Dataset $D = \{x_1, \dots, x_n\}$ with training categories $c_1, \dots, c_j$ and sum of training sample is $N$

1st step:
Repeat
        for each input vector $(x_1, \dots, x_n)$,
        do
        Calculate the similarities between training samples.
        e.g. ith sample of $d_i(d_{i1}, \dots, d_{im})$ where the similarity SIM$(x, d_i)$ can be represented by,
                $SIM(x, d_i) = \frac{\sum_{j=1}^{m} x_j . d_{ij}}{\sqrt{(\sum_{j=1}^{m} x_j)^2 \sqrt{\sum_{j=1}^{m} d_{ij}}^2}}$
2nd step:
Choose $k$ samples which are larger from $N$ similarities of $SIM(x, d_i)$, $(i = 1,2, \dots, N)$ and serve them
as KNN collection of $x$.

3rd step:
Calculate the probability of $x$ belong to different categories accordingly using,
                $P(x, c_j) = \sum_{d} SIM(x, d_i) . y(d_i . c_j)$
Where $y(d_i . c_j)$ defined by category attribute function that satisfied,
                $y(d_i . c_j) = \begin{cases} 1, d_i \in c_j \\ 0, d_i \notin c_j \end{cases}$

4th step:
Find the category for the sample that has the largest $P(x, c_j)$
```

Fig. 2: K-Nearest Neighbor (KNN) Algorithm

2.1.3. Decision Tree

Decision Tree is a model with flowchart-like structure. It is created by a tree and a set of rules representing each of the classes from a dataset. Decision Tree consists of three main elements which are internal node, branch and class label where each of them represents a test attribute, a test outcome and a leaf node respectively [13]. Fig. 3 shows the algorithm for Decision Tree [9].

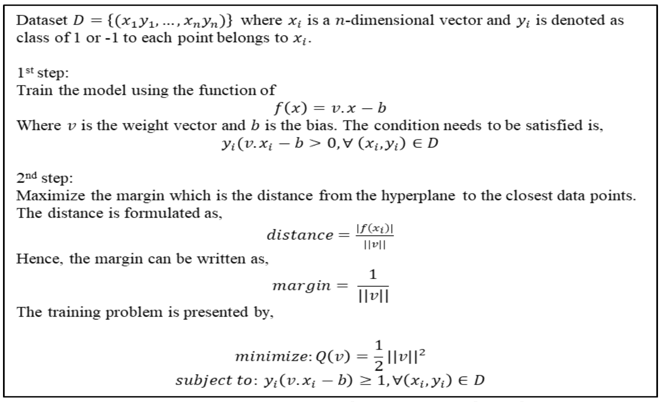


```
$D$ = Training Dataset, $M$ = Input Attributes and $N$ = Target Attributes

1st step:
Create a tree, $T = TreeGrowing(D, M, N)$
        If
        one of the stopping criteria is fulfilled then mark the root node in $T$ as a leaf
        with the most common value of $N$ in $D$ as the class.
        else
        Find a discrete function $f(M)$ of the input attributes values such that splitting
        $D$ based on outcome of $f(M)$ that gains the best splitting metric.
            If the best splitting metric ≥ threshold then label the root node of $T$ as $f(M)$
                For each outcome $v_i$ of $f(M)$ do
                $subtree_i = TreeGrowing\ (\sigma_{f(M)=v_i} D, M, N)$
                Connect the root node of $T$ to $subtree_i$ with an edge that is labelled $v_i$
                end for
            else
            Mark the root node in $T$ as a leaf with the most common value of $M$ in $D$ as
            the class.
            end if
        end if
The return output is the value of $T$

2nd step:
Prune the tree, $T = TreePruning(D, T, N)$
        Select a node $t$ in $T$ and prune it maximally improve some evaluation criteria
        If $t \neq \phi$ then
        $T = pruned(T, t)$
        end if
Repeat until $t = \phi$
The return output is the value of the current $T$
```

Fig. 3: Decision Tree Algorithm

2.1.4. Support Vector Machine

Support Vector Machine (SVM) is usually used for classification and was introduced by reference [14]. It works based on the calculation of margins between the classes. The margins are drawn to minimize the classification error when the distance between the margin and the classes is a maximum. SVM had been applied into various fields of studies such as gene expression, text classification and image identification [15]. This model is considered to give good generalization accuracy but it may cause a quadratic optimization problem with bound constraints and a lack of linear equality in the training process. The algorithm for SVM [16] is given in Fig. 4.

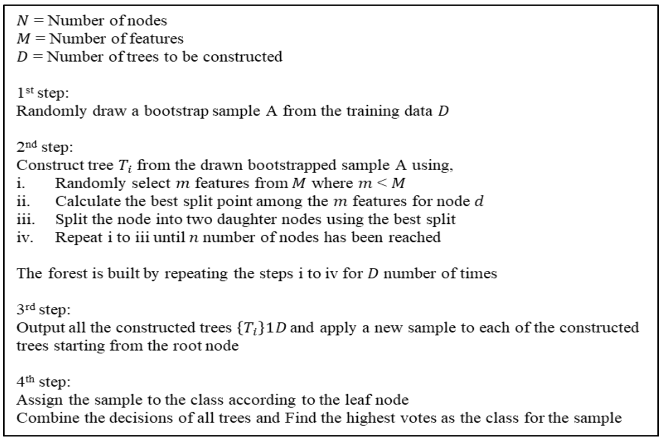


```
Dataset $D = \{(x_1 y_1, \ldots, x_n y_n)\}$ where $x_i$ is a $n$-dimensional vector and $y_i$ is denoted as
class of 1 or -1 to each point belongs to $x_i$.

1st step:
Train the model using the function of
                    $f(x) = v.x - b$
Where $v$ is the weight vector and $b$ is the bias. The condition needs to be satisfied is,
                    $y_i(v.x_i - b > 0, \forall\ (x_i, y_i) \in D$

2nd step:
Maximize the margin which is the distance from the hyperplane to the closest data points.
The distance is formulated as,
                    $distance = \frac{|f(x_i)|}{||v||}$
Hence, the margin can be written as,
                    $margin = \frac{1}{||v||}$
The training problem is presented by,

                    $minimize\colon Q(v) = \frac{1}{2}||v||^2$
                    $subject\ to\colon y_i(v.x_i - b) \geq 1, \forall (x_i, y_i) \in D$
```

Fig. 4: Support Vector Machine (SVM) Algorithm

2.1.5. Random Forest

Random Forest is also known as the ensemble of decision tree algorithm. Fig. 5 shows the algorithm for Random Forest [17]. It consists of a collection of tree-structured classifiers where each of the classifiers is an independent identically distributed random vector. This algorithm can maintain its performance even though the data consists of a large proportion of missing values [18].



```
$N$ = Number of nodes
$M$ = Number of features
$D$ = Number of trees to be constructed

1st step:
Randomly draw a bootstrap sample A from the training data $D$

2nd step:
Construct tree $T_i$ from the drawn bootstrapped sample A using,
i.    Randomly select $m$ features from $M$ where $m < M$
ii.   Calculate the best split point among the $m$ features for node $d$
iii.  Split the node into two daughter nodes using the best split
iv.   Repeat i to iii until $n$ number of nodes has been reached

The forest is built by repeating the steps i to iv for $D$ number of times

3rd step:
Output all the constructed trees $\{T_i\}1D$ and apply a new sample to each of the constructed
trees starting from the root node

4th step:
Assign the sample to the class according to the leaf node
Combine the decisions of all trees and Find the highest votes as the class for the sample
```

Fig. 5: Random Forest Algorithm

2.2. Dataset

Department of Statistics Malaysia (DOSM) has collected product information from one of the online store website in Malaysia through STATSBDA project known as Price Intelligence (PI) using its prototype web scraper. A few leaf nodes were used to represent the chosen categories from the browse tree of the website. Table 1 shows the description of the two corpora selected for this research which are fresh food and household products data. The six categories under Household data set are air freshener, floor cleaners, laundry, light bulbs, household sundries and toilet cleaner. Meanwhile, the five categories under Fresh Food category are bakery, fish & seafood, fresh fruits, fresh meat & poultry and noodles.

Table 1: Summary description of data sets

Dataset	Category	Instance	Number of Feature	Number of Feature after Feature Selection
Household	6	684	138	116
Fresh Food	5	447	88	78

2.3. Research Design

In this research, there were several steps involved before classifying the data as shown in Figure 6. The steps were data extraction, data pre-processing, feature extraction and feature selection. These were the basic steps in research related to text mining. After data extraction, there were three preprocessing steps involved which were tokenization, stop word removal and stemming [9]. The data preprocessing is a crucial step to ensure the data is standardized and in a proper form. The standardized form was achieved after applying the three preprocessing steps where product descriptions were tokenized into words at first. Then, stop words were removed from the word list and the remaining words were stemmed to ensure the words followed the root word forms.

The feature extraction and selection are important to make sure the data are well transformed into significance and good features before performing the classification process [19]. The selection of features may affect the accuracy of a classification model. Hence, the research had utilized the bag-of-word and correlation feature selection technique to perform data extraction and selection respectively. Then, the chosen features were used as inputs to perform different classification models from supervised learning models. All the steps were computed using R-Programming software.

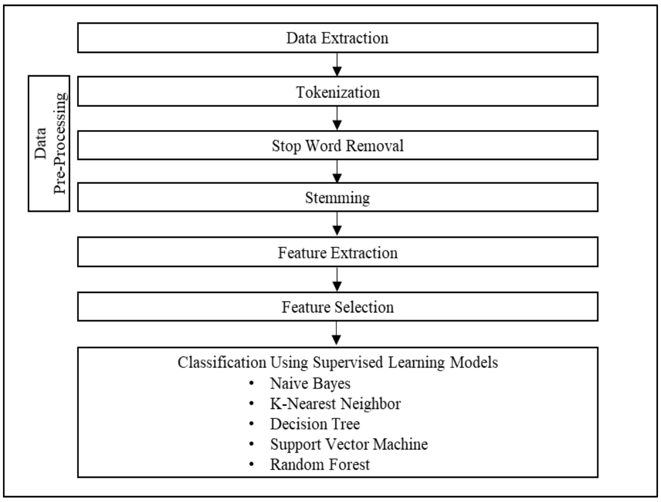


Fig 6: Flowchart of the Research

3. Results and Discussion

The evaluation was done by observing the classification results of five algorithms from supervised learning model. The analysis was made on two different datasets as mentioned in section 2.3. Table 2 shows the accuracy of classification models for household data set. Firstly, the highest accuracy for the data with six categories was performed by KNN model. On the other hand, the performance of Random Forest model was highly good as KNN model, but the performance of other classification models was more than 65% except for Naïve Bayes model. Besides that, the result from Fresh Food data led to approximately similar conclusion as the result obtained from Household data. The highest accuracy rate to classify the data consisted of five categories is KNN model. Specifically, only KNN and Random Forest models showed good accuracy rates compared to other classification models. The Naïve Bayes model was the worst classifier among the five algorithms used in the study to classify both of the data.

Method	Dataset	
	Household	Fresh Food
Naïve Bayes	16.99	14.07
KNN	94.66	82.96
Decision Tree	85.44	67.41
SVM	69.42	44.44
Random Forest	93.69	78.52

From the results, it was clear that KNN model outperformed other supervised learning models. However, the performance of Random Forest model was not far behind the KNN model. This result tied well with previous study by reference [20] wherein the performance of both models were preferable compared to other supervised learning models toward breast cancer data. Meanwhile, several studies had also found that KNN model is superior in classifying different kind of data [21]–[23]. Among the algorithms based on supervised learning models used in the study, Naïve Bayes performed not as good as the other algorithms. It is proved that the performance of Naïve Bayes model is affected by the distribution of the data [24]. Normally, it performed well on the real world data where the nature of the data drifts over the time. However, the data used in the study were independent and identically distributed data.

4. Conclusion

The paper presents comparative evaluation of different algorithms from supervised learning model for the problem related to classification of e-commerce products. Overall, KNN model performed the best compared to the other four classification models. For future work, the optimal number of neighbors (K) value of KNN model can also be investigated in enhancing the performance of the model. The performance of unsupervised and semi-supervised learning model can also be explored in classifying e-commerce products.

Acknowledgement

The research is financially supported by the University Teknologi MARA under the Grant Scheme 600-IRMI/PERDANA 5/3 BESTARI (062/2018). The authors would like to express their deepest gratitude to the Department of Statistics Malaysia for providing knowledge and data supports.

References

[1] D. Kim, S. Lee, and J. Chun, “A semantic classification model for e-catalogs,” *Proceedings. IEEE International Conference on e-Commerce Technology, 2004. CEC 2004.*, no. August, pp. 85–92, 2004.

[2] C. Sun, N. Rampalli, F. Yang, and A. Doan, “Chimera: Large-scale Classification Using Machine Learning, Rules, and Crowdsourcing,” *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1529–1540, 2014.

[3] R. Florez-Lopez and J. M. Ramon-Jeronimo, “Marketing segmentation through machine learning models: An approach based on customer relationship management and customer profitability accounting,” *Social Science Computer Review*, 2009.

[4] K. Balyan, K. S. McCarthy, and D. S. McNamara, “Combining machine learning and natural language processing to assess literacy text comprehension,” in *Proceedings of the 10th International Conference on Educational Data Mining (EDM)*, 2017.

[5] M. W. Libbrecht and W. S. Noble, “Machine learning applications in genetics and genomics,” *Nature Reviews Genetics*. 2015.

[6] P. Sharma and M. Kaur, “Classification in Pattern Recognition: A Review,” *International Journal of Advanced Research in Computer Science and Software Engineering*, 2013.

[7] R. Sathya and A. Abraham, “Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification,” *International Journal of advanced Research in Artificial Intelligence*, vol. 2, no. 2, pp. 34–38, 2013.

[8] S. Ryszard Michalski, G. Carbonell Jamie, and M. Tom Mitchell, *Machine learning: An Artificial Intelligence Approach*. Morgan Kaufmann, 1985.

[9] A. Dey, “Machine Learning Algorithms : A Review,” *International Journal of Computer Science and Information Technologies*, vol. 7, no. 3, pp. 1174–1179, 2016.

[10] A. S. Patil and B. V. Pawar, “Automated Classification of Web Sites using Naive Bayesian Algorithm,” *IMECS*, vol. 1, 2012.

[11] L. Devroye, “On the Inequality of Cover and Hart in Nearest Neighbor Discrimination,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1981.

[12] N. Suguna and K. Thanushkodi, “An Improved k-Nearest Neighbor Classification Using Genetic Algorithm,” *International Journal of Computer Science Issues*, 2010.

[13] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. 2012.

[14] C. Cortes and V. Vapnik, “Support-Vector Networks,” *Machine Learning*, 1995.

[15] C. D. A. Vanitha, D. Devaraj, and M. Venkatesulu, “Gene Expression Data Classification Using Support Vector Machine and Mutual Information-based Gene Selection,” *Procedia Computer Science*, 2015.

[16] H. Yu and S. Kim, “15 - SVM Tutorial — Classification, Regression and Ranking,” *Handbook of Natural Computing*, 2012.

[17] V. Y. Kulkarni and P. K. Sinha, “Effective Learning and Classification using Random Forest Algorithm,” *International Journal of Engineering and Innovative Technolgy*, vol. 3, no. 11, pp. 267–273, 2014.

[18] G. Krishna, M. Nookala, N. Orsu, B. K. Pottumuthu, and S. B. Mudunuri, “Performance Analysis and Evaluation of Different Data Mining Algorithms used for Cancer Classification,” *(IJARAI) International Journal of Advanced Research in Artificial Intelligence*, 2013.

[19] N. M. N. Mathivanan, N. A. M. Ghani, and R. M. Janor, “Improving Classification Accuracy Using Clustering Technique,” *Bulletin of Electrical Engineering and Informatics*, vol. 7, no. 3, pp. 465–470, 2018.

[20] G. N. Ramadevi, K. U. Rani, and D. Lavanya, “Evaluation of Classifiers Performance using Resampling on Breast Cancer Data,” *International Journal of Scientific & Engineering Research*, vol. 6, no. 2, 2015.

[21] X. Shao, H. Li, N. Wang, and Q. Zhang, “Comparison of different classification methods for analyzing electronic nose data to characterize sesame oils and blends,” *Sensors (Switzerland)*, 2015.

[22] D. R. Amancio *et al.*, “A systematic comparison of supervised classifiers,” *PLoS ONE*, 2014.

[23] P. Horton and K. Nakai, “Better prediction of protein cellular localization sites with the k nearest neighbors classifier.,” *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 1997.

[24] C. D. Manning, P. Ragahvan, and H. Schutze, *An Introduction to Information Retrieval*. Cambridge, England: Cambridge University Press, 2009.