Insolvency Prediction in Insurance Companies Using Support Vector Machines and Fuzzy Kernel C-Means

Zuherman Rustam¹, Frederica Yaurita²

¹,²Department of Mathematics, University of Indonesia, Depok, 16424, Indonesia
rustam@ui.ac.id

Abstract: Insolvency of insurance companies has been a concern of parties such as insurance regulators, investors, management, financial analysts, banks, auditors, policy holders, and consumers. This concern has arisen from the perceived need to protect the general public against the consequences of insurers insolvencies, as well as minimizing the responsibilities for management and auditors. In this paper we propose an approach to avoid insolvency in insurance companies. A large number of methods such as discriminant analysis, logit analysis, recursive partitioning algorithm, etc., have been used in the past for insolvency prediction. However, the special characteristics of the insurance sector have made most of them unfeasible, and just a few have been applied to this sector. In this study we predict the insololvency using two different methods, there are Support Vector Machines (SVM) and Fuzzy Kernel C-Means (FKCM). The results are very encouraging and show that SVM and FKCM can be a useful tool for parties who are interest in evaluating insololvency of an insurance firm.

1. Introduction
The main purpose of insurance regulation is to assure that insurers remain financially solvent and thus able to fulfill their contractual promises to their insureds. Based on PACIC research in 1995, the main reason of insurance insolvency was deficient loss reserves, while other reasons are because the failure of a foreign parent (home office), rapid growth, alleged fraud together, and overstated assets [1]. Early warning systems of financial failure is very important for insurance companies before it is too late to resolve problem insurers.

When the insurance company becomes financially insolvent there are adverse consequences for its stakeholder groups, such as insurance regulators, investors, management, financial analysts, banks, auditors, policy holders, and consumers, which could be impact into other firms, the wider economy and society [2]. This urgency made us interested to construct an efficient insolvency prediction model.

In this study, we used machine learning techniques to predict the insolvency in insurance companies. Machine learning is a method for deriving hypotheses from training data. This tool works by generalize and not to memorize instances in order to evaluate unseen data. Thus, we do not need to modify the program to apply it in different sectors. Machine learners are implemented in computer programs. In general, computer programs have the advantage of objective decision-making of the provided data not influenced by human opinions, because humans may also include their subjectivity to a certain degree.

Support Vector Machines (SVM) and Fuzzy Kernel C-Means (FKCM) are known as the powerful machine learning tool for classification. Application of SVM has been used by I.P.A Wardana and Z. Rustam [3], D.A. Puspitasari and Z. Rustam [4], Z. Rustam, D.F. Vibranti, and D. Widya [5] for
decision-making in stock investment, stock analysis of Indonesian stock exchange, and predicting the direction of Indonesian stock price movement respectively, while FKCM has been used by Z. Rustam, D.F. Vibranti and D. Widya [5], Z. Rustam and Fanita [6], P. Hulyani Kin and Z. Rustam [7] for predicting the direction of Indonesian stock price movement, predicting the composite index price, and forecasting stock market momentum respectively.

Because a lot of financial problem could be solved by SVM and FKCM then the results are trustworthy, so in this study we try to conduct the insolvency prediction model used both algorithms. We built the model by using several insurance firms data which got from Prof. Dr. Maria Jesus Segovia. The results are very encouraging and show that SVM and FKCM can be a useful tool to avoid the insolvency.

The rest of paper is structured as follows. In the next section, there are brief introduction to Support Vector Machines and Fuzzy Kernel C-Means. The third section describes the classifier performance that we have been used (confusion matrix and area under the curve). The forth section includes the analysis of the test data used, the experiments performed and the result obtained. Finally, in the last section some concluding remarks can be found.

2. Theoretical Background

2.1. Support Vector Machines (SVM)

SVM has received much attention in classification problems. This method first proposed by Vapnik in 1998. In a few years later, Nello Cristianini (Professor of Artificial Intelligence at the University of Bristol) did a research about SVM based on Vapnik’s theory before, and have written the result in [8]. After that in 2002 Bernhard Scholkopf developed SVM theory and published it in a book which is about kernels in SVM as we can see in [9]. We used [8], [9] and [10] as our main reference in this SVM section.

Let a set of firms represented by the value of their ratios $x_i, i = 1, ..., N$, and a set of associated labels $y_i \in \{-1, 1\}$ which describe the firm as failed or healthy. The main purpose of SVM is find the best hyperplane:

$$w \cdot x + b = 0 \quad (1)$$

which able to maximize the margins.

The optimization problem of SVM can be summarized as:

$$\text{Minimize} \quad \frac{1}{2} ||w||^2 \quad (2)$$

subject to:

$$y_i(w^T \cdot x_i + b) \geq 1, \quad \forall i = 1, ..., N \quad (3)$$

The objective of (2) is to find $w \in \mathbb{R}^n$ and $b \in \mathbb{R}$ with constraints (3), where $w$ is set of weights and $b$ is the bias.

Problem (2) is a quadratic optimization problem. We introduce Lagrange multipliers $\alpha_i$, for each of the constraints in (2), giving the Lagrangian function:

$$L(w, b, \alpha) = \frac{1}{2} ||w||^2 - \sum_{i=1}^{N} \alpha_i \{y_i(w \cdot x_i + b) - 1\} \quad (4)$$

where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)^T$. Setting the derivatives of $L(w, b, \alpha)$ with respect to $w$ and $b$ equal to zero, the following two conditions were obtained:

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{N} a_i y_i x_i = 0 \Rightarrow w = \sum_{i=1}^{N} a_i y_i x_i \quad (5)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{N} a_i y_i = 0 \Rightarrow \sum_{i=1}^{N} a_i y_i = 0 \quad (6)$$

By eliminating $w$ and $b$ from $L(w, b, \alpha)$ using (5) and (6) conditions, we obtained the dual form:
\[ L(\alpha) = \max \left\{ -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j a_i a_j (x_i, x_j) + \sum_{i=1}^{N} a_i \right\} \]  

subject to:

\[ \sum_{i=1}^{N} y_i a_i = 0, \quad a_i \geq 0 \]  

(7)

There is a case that the dataset is not linearly separable, therefore dataset can be transformed into multi-dimension using kernel function \( K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \). In this paper, we used RBF kernel as described in Eq. (20), and by changing the inner product of two vectors with \( K(x_i, x_j) \), Eq. (7) can be rewritten as:

\[ L(\alpha) = \max \left\{ -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j a_i a_j K(x_i, x_j) + \sum_{i=1}^{N} a_i \right\} \]  

(9)

Solving Eq. (9) with constraints in Eq. (8) determines the Lagrange multipliers \((a_i, a_j)\) and the \( w \) and \( b \) of regression function given by \( f(x) = w \cdot x + b \) are finally obtained as follows:

\[ w = \sum_{i=1}^{N} a_i y_i x_i \]  

(10)

thus,

\[ f(x) = \sum_{i=1}^{N} a_i y_i x_i + b \]  

(11)

and

\[ b = \frac{1}{N} \sum_{i \in S} \left( y_i - \sum_{m \in S} a_m y_m x_m \right) \]  

(12)

### 2.2. Fuzzy Kernel C-Means (FKCM)

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

FCM or Fuzzy C-Means method is one of fuzzy clustering method [11]. For a data set \( X = \{x_1, x_2, ..., x_m\} \subseteq \mathbb{R}^d \), we define \( n \times c \) Membership Matrix \( U = [u_{ij}], 1 \leq i \leq n, 1 \leq j \leq c \), and Cluster Center \( V = \{v_1, v_2, ..., v_c\} \) where each object in \( V \) is an element of \( d \)-dimensional Euclidean Space [12].

Mathematical model of Fuzzy C-Means can be expressed as:

\[ f(U, V) = \min \sum_{i=1}^{n} \sum_{j=1}^{c} (u_{ij})^m d^2(x_i, v_j) \]  

with constraints:

\[ \sum_{j=1}^{c} u_{ij} = 1, \quad i = 1, 2, ..., n \]

\[ \sum_{i=1}^{n} u_{ij} > 0, \quad j = 1, 2, ..., c \]  

\[ u_{ij} \in [0, 1], \quad j = 1, 2, ..., c \]  

(13)

(14)
where \( d \) is distance or dissimilarity function, and \( m \in [1, \infty) \) is the fuzziness degree for cluster partition.

Cluster center and membership values are updated by using:

\[
y_j = \frac{\sum_{i=1}^{n} u_{ij}^m x_i}{\sum_{i=1}^{n} u_{ij}^m}, j = 1, 2, ..., c
\]

and

\[
u_{ij} = \left( \sum_{j=1}^{c} \left( \frac{d(x_i, v_j)}{d(x_i, v_j)} \right)^{\frac{2}{m-1}} \right)^{-1}, 1 \leq i \leq n
\]

FCM classification’s accuracy is dependent on the types of data. When the data is non-linearly separable, its convergence is slow and not accurate. To solve this problem, the data set are transformed into another space (feature space) that dimension is much higher than the data space [12]. It is expected that the transformed data behavior can approach the linearly separable data, so that classification accuracy can be improved.

Similar with SVM method, while working directly on feature space (high dimensional space), we need another tool as a “connector” between data space and feature space, so we can have a better accuracy without directly working at feature space [12]. This concept is called Kernel method that was given by Vapnik [13] and elaborate further by Platt [14], Christianini and Taylor [15], and Scholkopf et al. [16].

Set a nonlinear mapping \( \varphi \) from input data space \( \mathbb{R}^d \) into feature space \( F \). The clustering process take place at \( F \) other than \( \mathbb{R}^d \). We need to find a way to measure distance between transformed data \( \varphi(x) \) and \( \varphi(y) \), \( x, y \) are objects at data space without knowing explicit form of \( \varphi \) [12]. To solve this problem we use kernel function \( K \) as in [13]. By using kernel function, distance between \( \varphi(x) \) and \( \varphi(y) \) can be measure by:

\[
d^2(\varphi(x), \varphi(y)) = ||\varphi(x) - \varphi(y)||^2
\]

\[
= \varphi(x)^t \varphi(x) - 2\varphi(x)^t \varphi(y) + \varphi(y)^t \varphi(y)
\]

\[
= k(x, x) - 2k(x, y) + k(y, y)
\]

(17)

The success of kernel method on classification problem [15, 16] had inspired other researchers to apply kernel method on classic classification method, as in [17] that combine Fuzzy K-Medoids algorithm and kernel method to solve multiclass multidimensional data classification problem.

At this paper we use Fuzzy Kernel C-Means (FKCM) that comes by applying kernel method into FCM method to solve insolvency prediction problem.

In Karayiannis and Bezdek research about Fuzzy LVQ [18], for each iteration, a different fuzziness degree \( m \) are used:

\[
m = m_i + \frac{r}{T} (m_f - m_i)
\]

(18)

where \( m_i \) and \( m_f \) are initial value and end value of \( m \), respectively. When the value of \( m_f \) is small and \( m_i \) is quite big then it is expected that \( m \) will decreasing and vice versa [12].

Fuzzy Kernel C-Means Algorithm (FKCM) was built by applying kernel method into FCM, while using [18] concept to choose fuzziness degree.
Figure.1. Fuzzy Kernel C-Means Algorithm
According to Bezdek [5], sequences \( \{U^t, V^t\} \) will converge to minimum value of \( f(U, V) \).

3. Classifier Performance
Confusion matrix has received much attention for determine a classifier performance. A confusion matrix shows the number of correct and incorrect predictions made by the classification model compared to the actual outcomes in the data. Table.1. shows the general form of confusion matrix

| Observed | Predicted |
|----------|-----------|
| Healthy  | Correct Healthy | Type II Error |
| Failed   | Type I Error | Correct Failed |

| Definition | True Positives (TP) | The number of healthy firms that were classified healthy. |
|------------|---------------------|--------------------------------------------------------|
| True Negatives (TN) | The number of failed firms that were classified failed. |
| False Positives (FP) | The number of failed firms that were classified healthy. |
| False Negatives (FN) | The number of healthy firms that were classified failed. |

The accuracy is the proportion of the total number of predictions that were correct. It is determined using the equation:

\[
\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{19}
\]

The higher accuracy shows the better classifier performance.
4. Experimental Result

4.1. Data and Variables

In this study, we built the model by using Spanish non-life insurance firms data which is got from Prof. Dr. Maria Jesus Segovia [2]. In prior research, she built insolvency prediction model used Rough Sets method as we can see in [2].

In this experiment, there are 72 firms selected, it consists of 36 failed firms and 36 healthy firms. As a control measure, a failed firm is matched with a healthy one in terms of industry and size. In this data set, each firm is described by 21 ratios. Table.3. shows all of ratios definition that we have used. Further information about the ratios can be check in [2].

| Ratio | Definition |
|-------|------------|
| R1    | Working Capital/ Total Assets |
| R2    | Earning Before Taxes (EBT)/(Capital + Reserves) |
| R3    | Investment Income/ Investments |
| R4    | EBT*/ Total Liabilities  
    | EBT* = EBT + Reserves for Depreciation + Provisions +  
    | (Extraordinary Income - Extraordinary Charges) |
| R5    | Earned Premiums/ (Capital + Reserves) |
| R6    | Earned Premiums Net of Reinsurance/ (Capital + Reserves) |
| R7    | Earned Premiums/ (Capital + Reserves + Technical Provisions) |
| R8    | Earned Premiums Net of Reinsurance/ (Capital + Reserves + Technical Provisions) |
| R9    | (Capital + Reserves)/ Total Liabilities |
| R10   | Technical Provisions/ (Capital + Reserves) |
| R11   | Claims Incurred/ (Capital + Reserves) |
| R12   | Claims Incurred Net of Reinsurance/ (Capital + Reserves) |
| R13   | Claims Incurred/ (Capital + Reserves + Technical Provisions) |
| R14   | Claims Incurred Net of Reinsurance/ (Capital + Reserves + Technical Provisions) |
| R15   | Combined Ratio 1 = (Claims Incurred/ Earned Premiums) + (Other Charges and Commissions/ Other Income) |
| R16   | Combined Ratio 2 = (Claims Incurred Net of Reinsurance/ 
    | Earned Premiums Net of Reinsurance) + (Other Charges and Commissions/ Other Income) |
| R17   | (Claims Incurred + Other charges and Commissions)/ Earned Premiums |
| R18   | (Claims Incurred Net of Reinsurance + Other charges and Commissions)/ Earned Premiums Net of Reinsurance |
| R19   | Technical Provisions of Assigned Reinsurance/ 
    | Technical Provisions |
| R20   | Claims Incurred/ Earned Premiums |
| R21   | Claims Incurred Net of Reinsurance/ 
    | Earned Premiums Net of Reinsurance |

All ratios are provided in continuous variables and we have recoded it into qualitative terms (low, medium, high, and very high) with corresponding numerical values such as 1, 2, 3 and 4. That recoding made by dividing the original domain into four subintervals. In order to find the best model we did several experiments, we used those two different input data types/regarding model inputs.

In machine learning, before we use the algorithm, we should split the data set into training set and testing set. Training set is implemented to build up a model while the testing set is to validate the model built (check the accuracy). Based on the prior experiments, most of training set are not linearly separable. To solve this problem, the data set should be transformed into another space (feature space) that dimension is much higher than the data space. It is expected that the transformed data behavior
can approach the linearly separable data, so that classification accuracy can be improved. In this experiment, we used RBF kernel which defined in (20).

$$k(x_i, x_j) = \exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$$  \hspace{1cm} (20)

4.2. Laplacian Score as Feature Selection

A lot of classical feature selection methods have been proposed. Feature selection is use to eliminates irrelevant and redundant ratios of every firm and regularly improves the model performance. The remaining ratios are used by the SVM and FKCM for the classification process. In this study we used Laplacian Score. The objective function of Laplacian Score is defined as follows:

$$L_r^{LS} = \frac{\sum_{ij}(f_{ri} - f_{rj})^2 S_{ij}}{\text{Var}(f_r)}$$ \hspace{1cm} (21)

where $L_r^{LS}$ is the Laplacian score of the r-th feature, $f_{ri}$ and $f_{rj}$ denote the r-th feature of the samples $x_i$ and $x_j$. Var($f_r$) is the estimated variance of the r-th feature. $S$ is the unsupervised weight matrix of the k nearest neighbour graph, which is defined as

$$S_{ij} = \exp\left(-\frac{||x_i - x_j||^2}{t}\right)$$ \hspace{1cm} (22)

if $x_i$ and $x_j$ are k nearest neighbors, and $S_{ij} = 0$ for otherwise. Take a look at [19] for more information about this feature selection.

4.3. Results

Table.IV. and Table.V. show the summary of our experiments, respectively for continuous and discrete input type. In this experiment we used RBF kernel function and 0.05 parameter. As we can see from both of table, SVM and FKCM are split into two categories, those are experiment without using feature selection (using all features) and experiment with using feature selection. Note that the best solution achieved by Laplacian Score involves 9 ratios R1, R3, R4, R7, R8, R9, R13, R14, R17.

Table.4. Accuracy and running time of svm and fkcm method with continuous input data types using rbf kernel function and $\sigma = 0.05$

| Training Data (%) | Continuous | Support Vector Machines | Fuzzy Kernel C-Means |
|-------------------|------------|-------------------------|----------------------|
|                   | All Features | Feature Selection | All Features | Feature Selection |
|                   | Accuracy | Running Time | Accuracy | Running Time | Accuracy | Running Time | Accuracy | Running Time |
| 10                | 60.94   | 3.39    | 62.50   | 0.14    | 48.44   | 0.45    | 60.94   | 0.13    |
| 20                | 67.86   | 3.39    | 44.64   | 0.08    | 53.57   | 0.81    | 53.57   | 0.09    |
| 30                | 64.00   | 3.42    | 42.00   | 0.14    | 46.00   | 1.03    | 44.00   | 0.11    |
| 40                | 71.43   | 3.42    | 54.76   | 0.13    | 54.76   | 0.25    | 45.24   | 0.19    |
| 50                | 61.11   | 4.69    | 61.11   | 0.08    | 55.56   | 0.30    | 69.44   | 0.08    |
| 60                | 53.57   | 4.98    | 67.86   | 0.16    | 71.43   | 0.61    | 71.43   | 0.09    |
| 70                | 75.00   | 5.30    | 60.00   | 0.11    | 75.00   | 2.36    | 70.00   | 0.42    |
| 80                | 50.00   | 5.48    | 64.29   | 0.17    | 85.71   | 2.61    | 71.43   | 0.56    |
| 90                | 16.67   | 5.78    | 83.33   | 1.03    | 100.00  | 2.98    | 100.00  | 0.31    |
Table.5. Accuracy and running time of svm and fkcm method with discrete input data types using rbf kernel function and $\sigma = 0.05$

| Training Data (%) | Support Vector Machines | Fuzzy Kernel C-Means |
|-------------------|-------------------------|----------------------|
|                   | All Features            | Feature Selection    | All Features | Feature Selection |
|                   | Accuracy Running Time   | Accuracy Running Time | Accuracy     | Running Time      |
| 10                | 62.50 0.14              | 70.31 0.08           | 68.75 0.08   | 56.25 0.38       |
| 20                | 44.64 0.08              | 71.43 0.02           | 69.64 0.05   | 58.93 0.61       |
| 30                | 42.00 0.14              | 68.00 0.05           | 64.00 0.02   | 60.00 0.75       |
| 40                | 54.76 0.13              | 66.67 0.00           | 59.52 0.02   | 61.90 0.97       |
| 50                | 61.11 0.08              | 66.67 0.03           | 63.89 0.00   | 66.67 1.13       |
| 60                | 67.86 0.16              | 67.86 0.02           | 75.00 0.03   | 71.43 1.38       |
| 70                | 60.00 0.11              | 65.00 0.00           | 70.00 0.00   | 75.00 1.66       |
| 80                | 64.29 0.17              | 71.43 0.00           | 71.43 0.03   | 71.43 1.83       |
| 90                | 83.33 1.03              | 100.00 0.03          | 100.00 0.00  | 100.00 2.08      |

Table.6. Confusion Matrix Of Svm With Continuous Input Data Types Using All Features And 90% Training Data

| Predicted | Failed | Healthy |
|-----------|--------|---------|
| Observed  |        |         |
| Failed    | 50     | 0       |
| Healthy   | 17     | 33      |

Table.7. Confusion Matrix Of Svm With Discrete Input Data Types Using All Features And 90% Training Data

| Predicted | Failed | Healthy |
|-----------|--------|---------|
| Observed  |        |         |
| Failed    | 50     | 0       |
| Healthy   | 0      | 50      |

It can be seen from Table.IV, that the best model of SVM with continuous type is obtained by using feature selection, the average accuracy is 60.05%. While the best model of FKCM with continuous type is obtained when we used all features, the average accuracy is 65.61%. Furthermore, from Table.V., we can see that the best model of SVM with discrete type is obtained by using feature selection, it is 71.93% average accuracy. FKCM with discrete type obtained its best average accuracy when we conducted the model without feature selection, it is 71.36% average accuracy.

Table.VI. and Table.VII. show the confusion matrix of SVM with discrete input data types, respectively using all features and feature selection.

5. Conclusions
Based on analysis of the experimental results above, it can be concluded that the use of discrete type of input have a significant effect on both SVM and FKCM method. The highest average accuracy is obtained by SVM with discrete input data types using feature selection, it is 71.93%. In future studies, we interested to build the insolvency prediction model using Indonesian Insurance data set.

References
[1] Massart Thierry, Meuter Cedric, V B Laurent. On the complexity of partial order trace model checking. Inform. Process. Lett. 106 (2008) 120-126

[2] S. Dibra, D. Leadbetter. (2007). Why Insurance Fail: The dynamics of property and casualty insurance insolvency in Canada. Canada: Property and Casualty Insurance Compensation Corporation.
[3] M.J. Segovia-Vargas, J.A. Gil. Fana, A. Heras-Martinez, J.L. Vilar-Zanon, A. Sanchis-Arellano. (2002). Using Rough Sets to predict insolvency of Spanish non-life insurance companies. Madrid.

[4] I.P.A Wardana, Z. Rustam. (2017). Application of Support Vector Machines on Decision-Making in Stock Investment. Proceeding of 3rd International Symposium on Current Progress in Mathematics and Sciences.

[5] D.A. Puspitasari, Z. Rustam. (2017). Application of SVM-KNN using SVR as Feature Selection on Stock Analysis for Indonesian Stock Exchange. Proceeding of 3rd International Symposium on Current Progress in Mathematics and Sciences.

[6] Z. Rustam, D.F. Vibranti, D. Widya. (2017). Predicting The Direction of Indonesian Stock Price Movement using Support Vector Machines and Fuzzy Kernel C-Means. Proceeding of 3rd International Symposium on Current Progress in Mathematics and Sciences.

[7] Z. Rustam, Fanita. (2017). Predicting The Jakarta Composite Index Price using ANFIS and Classifying Prediction Result Based on Relative Error by Fuzzy Kernel C-Means. Proceeding of 3rd International Symposium on Current Progress in Mathematics and Sciences.

[8] P. Hulyani Kin, R. Zuherman. (2016) A Fuzzy Logic Model to Forecast Stock Market Momentum in Indonesia’s Property and Real Estate Sector. International Symposium on Current Progress in Mathematics and Sciences.

[9] N. Christianini and J.S. Taylor. (1999). An Introduction to Support Vector Machines and Other Kernel based Learning Methods, Cambridge.

[10] Scholkopf, B., & Smola, A. J. (2002). Learning with kernels: Support vector machines, regularization, optimization, and beyond. Cambridge: MIT press.

[11] C.M. Bishop. (2006). Pattern Recognition and Machine Learning. UK: Springer.

[12] J. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms, New York, Plenum Press, 1981

[13] Z. Rustam, & Suri, Aini. (2015). Fuzzy Kernel C-Means Algorithm For Intrusion Detection Systems. Journal of Theoretical and Applied Information Technology, Vol. 81, Issue 1, November 2015.

[14] V. N. Vapnik, The Nature of Statistical Learning Theory, New York, Springer-Verlag, 2000.

[15] J. C. Platt, Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In: Advances in Kernel Methods – Support Vector Learning. MIT Press Cambridge, 1999.

[16] N. Christianini and J.S. Taylor, An Introduction to Support Vector Machines and Other Kernel-based Learning Methods, Cambridge University Press, 2000.

[17] B. Scholkopf, A. Smola, and K. R. Muller, “Nonlinear Component Analysis as a Kernel Eigenvalue Problem”, Neural Computation, Vol. 10, No. 5, 1998, pp. 1299-1319.

[18] Z. Rustam and A. S. Talita, “Fuzzy Kernel K-Medoids Algorithm for Multiclass Multidimensional Data Classification”, Journal Multidimensional Data Classification”, Journal of Theoretical and Applied Information Technology, Vol. 80, Issue 1, October 2015.

[19] N. B. Karayiannis and J. C. Bezdek, “An Integrated Approach to Fuzzy Learning Vector Quantization and Fuzzy C-Means Clustering”, IEEE Trans. Fuzzy Systems, Vol. 5, No. 4, 1997, pp. 622-628.

[20] X.F. He, D. Cai, P. Niyogi. (2005). Laplacian score for feature selection, in: Advances in Neural Information Processing Systems. Cambridge.