International Journal of Technology (2016) 4: 709-719
ISSN 2086-9614

PERFORMANCE ANALYSIS OF AN AUTOMATIC GREEN PELLET NUCLEAR FUEL QUALITY CLASSIFICATION USING MODIFIED RADIAL BASIS FUNCTION NEURAL NETWORKS

Benyamin Kusumoputro[1*], Dede Sutarya [2], Akhmad Faqih[1]

[1] *Computational Intelligence and Intelligent System Research Group, Dept. of Electrical Engineering, Faculty of Engineering, Universitas Indonesia, Kampus Baru UI, Depok 16424, Indonesia*
[2] *Center of Technology for Nuclear Fuels, Badan Tenaga Atom Nasional, Kompleks PUSPIPTEK Serpong, Serpong 15314, Indonesia*

(Received: February 2016 / Revised: March 2016 / Accepted: April 2016)

ABSTRACT

Cylindrical uranium dioxide pellets, which are the main components for nuclear fuel elements in light water reactors, should have a high density profile, a uniform shape, and a minimum standard quality for their safe use as a reactor fuel component. The quality of green pellets is conventionally monitored by laboratory measurement of the physical pellet characteristics; however, this conventional classification method shows some drawbacks, such as difficult usage, low accuracy, and high time consumption. In addition, the method does not address the non-linearity and complexity of the relationship between pellet quality variables and pellet quality. This paper presents the development and application of a modified Radial Basis Function neural network (RBF NN) as an automatic classification system for green pellet quality. The weight initialization of the neural networks in this modified RBF NN is calculated through an orthogonal least squared method, and in conjunction with the use of a sigmoid activation function on its output neurons. Experimental data confirm that the developed modified RBF NN shows higher recognition capability when compared with that of the conventional RBF NNs. Further experimental results show that optimizing the quality classification problem space through eigen decomposition method provides a higher recognition rate with up to 98% accuracy.

Keywords: Green pellet quality classification; Nuclear fuel cell; Orthogonal least squared method; RBF NN; Weight initialization

1. INTRODUCTION

The growth in energy consumption predicted for Indonesia is about 7% per annum until 2025. Consequently, the Government of Indonesia is now planning to seek alternative energy sources to compensate for predicted energy shortages, in order to ensure the country's long term economic development. Fossil fuel has been the primary energy source for electrical power plants but will be insufficient to supply this increasing demand for energy. For this reason, the construction of light water reactor nuclear power plants is planned, and these are expected to be operational by 2020, contributing 4% of the demand for electricity (Briyatmoko et al., 2010). Therefore, the fabrication of nuclear fuel elements (i.e., cylindrical uranium dioxide pellets)

*Corresponding author's email: kusumo@ee.ui.ac.id, Tel. +62-21-7270078, Fax. +62-21-7270077

Permalink/DOI: http://dx.doi.org/10.14716/ijtech.v7i4.3138

should conform to a standard that delivers uniform shape and quality, with a high density profile (Pramanik et al., 2010). However, some of the green pellets produced from the Experimental Fuel Element Installation (IEBE), Serpong, Indonesia might not meet the required standards, and should be rejected, which would increase the fuel fabrication cost. Here, an automatic quality classification system using a neural network is proposed that would minimize the number of green pellets rejected at the end of the fabrication process.

Conventionally, the quality of the green pellets is monitored by a laboratory measurement, but this measurement has some drawbacks: it is difficult to use, gives a low accuracy classification, and is time consuming. The classification process plays an important role in determining the quality of the nuclear fuel elements; therefore, an automatic classification method that could provide a higher classification rate with lower processing time is desirable. In this paper, we developed an accurate pellet quality classification method based on Radial Basis Function Neural Networks (RBF NNs) by modifying the weight initialization of the networks (Kusumoputro et al., 2013) and using a sigmoid activation function on its output neurons. Weight initialization is very important in a neural network learning mechanism, since different initialization values lead to different directions in its gradient descent, which increases the error convergence and computational costs.

This paper is organized as follows. Section II presents a description of the materials and methods used in processing the green pellets and preparing the data collection. Section III discusses the design and the development of the neural network-based classifier, i.e. radial Basis Function, in detail, including with the proposed weight initialization method. The validity of the robustness of the proposed classifiers are verified by a comparison of the experimental results, which is presented in Section IV, follows by the conclusion presented in Section V.

2. MATERIALS AND METHODS

The materials used in this research were obtained from the Experimental Fuel Element Installation (IEBE) laboratory at Serpong, Indonesia. This laboratory processes the raw materials (i.e., “yellow cake”) into an uranium dioxide powder for nuclear energy and conducts further processing to form fuel bundles (Batan, 2007). The green pellets used in this research are fabricated by following the standard operating procedure defined by the IEBE Laboratory (Batan, 2007), including quality assessment through a conventional system that can be briefly explained as follows. After preparing and testing the quality of the powder, the powder is compacted after mixing it with lubricant ingredients. High density and defect free uranium dioxide fuel pellets are then produced through an ammonium-diuranate precipitate route. This procedure involves numerous steps, including pressure agglomeration of the uranium dioxide powder, admixing with the lubricant, die compaction, and a solid state sintering process in high temperature chamber. The result is a high percentage yield of high quality green pellets.

The output product of this fabrication process is then classified into three categories: Upper Quality (*UQ*), Standard Quality (*SQ*), and Low Quality (*LQ*). The standard optimal green pellet production processes should produce 150 pellets of high quality in one process lot (i.e. *SQ* and *UQ* only). Unfortunately, some of the pellets usually do not conform to the specification standards (i.e., they are *LQ*), and these need to be destroyed by acid dissolution and undergo the fabrication process again. The green pellet classification procedure is recognized as being crucial and important, because this process leads to rejection or acceptance of the green pellets for the next process. The conventional method for classification, however, is a human inspection system, which is prone to false classification and is time consuming. For these reasons, a total quality classification process is proposed that uses a computational intelligence method.

The parameters for classifying the quality of the green pellets are measured using equipment of high standard and include the pellet density (*D*), the theoretical density (*TD*), the pellet weight (*W*), the pellet volume (*V*), and the pellet height (*H*). The relationship between the pellet density (*D*) and the theoretical density (*TD*) also needs to be considered. The pellet density (*D*) should be in the range of 50 to 60%, which gives a theoretical density (*TD*) for the pellets after the sintering process in the range of 95 to 97% (Jayaraj & Ganguly, 2003). The precise arrangement of the green pellets in a bundle requires, for practical purposes, that the ratio of the pellet height to its diameter be in the range of 1.0 to 1.1.

The data used for determining the quality classification of the green pellets are measured three times and averaged to ensure validity. The training and testing of the neural networks developed in this research was conducted by dividing the total data obtained from one lot from the green pellet fabrication process into two datasets, i.e., a training dataset and a testing dataset with various different percentages. The characteristics of the quality parameter data used for classification of the green pellets are shown in Figure 1, which presents 20 random data items for the five classification parameters.

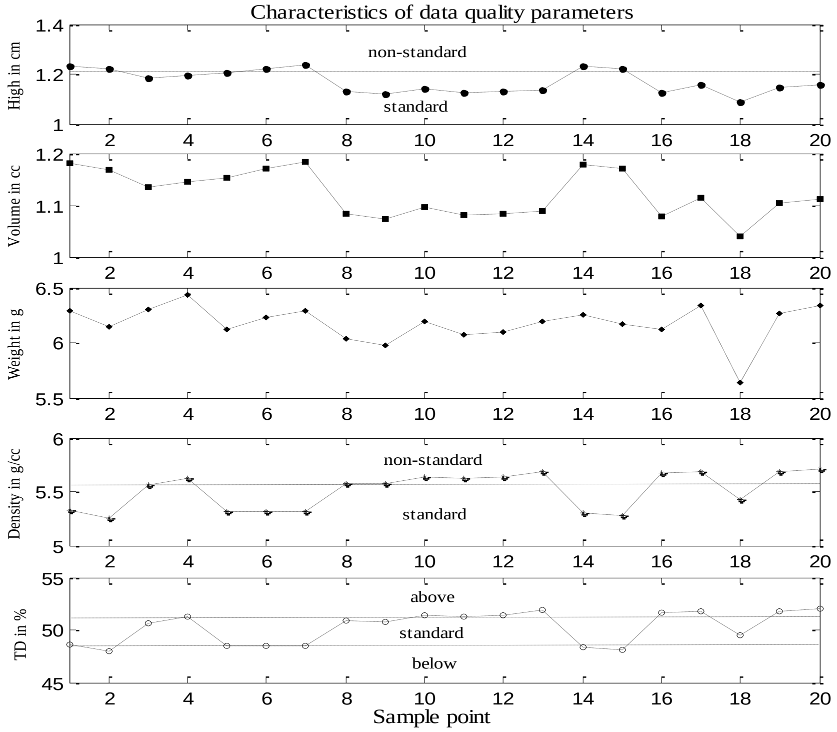

Figure 1 Sample of the data distribution as a function of five quality parameters of green pellets

3. MODIFIED RADIAL BASIS FUNCTION NEURAL NETWORKS

Artificial neural networks represent the massive parallel interconnections of a simple neuron that functions as a collection system, and they are designed in an attempt to mimic the human brain in order to emulate human performance, i.e., to function intelligently. Neural networks are now increasingly used to improve the recognition capability of intelligent systems because they show better performance in terms of simulating non-linear relationships, especially in problems that do not explicitly provide physical or mathematical representation, by imparting the characteristics of parallel computation, self-adapting learning, and better fault tolerance (Jothiprakash et al., 2009; Garcia et al., 2011; Sarma, 2009). The multilayer perceptron usually consists of two layers of simple two-stage sigmoid-processing artificial neurons that interact through weighted connections.

Training the neural networks with supervised learning algorithm results in a nonlinear mapping or an association task that can also be viewed as a curve-fitting process. Suppose the weight of a neural network is initialized by any random value and given a pair of two-set data; that is, input-output (as a target) data pairs. The neural network performs a weight adjustment, back and forth, to develop a specific nonlinear mapping according to the relation between the input-output data pairs. The weight adjustment is derived and calculated during the training phase, through the difference between the real output of the network and the target specified by the input-output data pairs, which is minimized over all input-output data pairs.

The generalization ability of a single neural network, however, depends on many parameters that are difficult to control for different applications in many real world problems. The use of a back propagation method as supervised learning, which is developed based on gradient descent of quadratic error function, had led to adoption of the powerful MLP (BP NN) as a universal approximator (Hornik et al., 2009). Back propagation based MLP (BP NN) can approximate any smooth function to an arbitrary degree of accuracy when the tuning parameters are properly optimized. However, the BP NN has a number of drawbacks, such as a high computational costs and the possibility of being trapped in local minima.

Another supervised learning method that has the same characteristics as BP NN but shows lower computational costs is the radial basis function neural network (RBF NN). Instead of using a conventional neuron calculation in a hidden neuron of the BP NN, the RBF NN uses a Gaussian function as its functional form for calculating the neuron activation in its hidden neuron; hence, it is called a Gaussian RBF NN. Similar to the BP NN architecture, the RBF NN consists of three layers, i.e., an input layer, a single hidden layer of nonlinear processing neurons, and an output layer.

For an input vector $\boldsymbol{x}^p \in \Re^{Nx1}$ with $p = 1, 2, ..., P$ as the number of the learning vector pairs, the output of the RBF NN is calculated according to:

$$y_k^p = \sum_{j=1}^{J} w_{jk}\phi_j\left(\| \boldsymbol{x}^p - \boldsymbol{c}_j \|\right) \tag{1}$$

where $\phi(\cdot)$ is the transfer function from $\Re^+$ (the set of all positive real numbers) to $\Re$, which is determined as a Gaussian function in this research; $\|\cdot\|$ denotes the Euclidean form, w_{jk} are the weights of the output layer with $j = 1, 2, ..., J$ as the number of neurons in the hidden layer, and $\boldsymbol{c}_j \in \Re^{Nx1}$ are the RBF centers that are usually determined in advance using various methods. As can be clearly seen from (1), for each incoming input vector to the neural network, the Euclidean distance is calculated between each neuron in the hidden layer and the input vector. Each hidden neuron, ϕ_j, therefore outputs a different value as a nonlinear function of this distance. The output of the networks y_k^p is then calculated as a total weighted sum through all the hidden layer outputs. The Gaussian function used in the developed method is determined as $\phi(x) = \exp(-\frac{x^2}{\sigma^2})$, where σ, as the spread parameter, is determined to be the same for all hidden neurons.

The curve-fitting characteristics of the RBF NN depends on three set of parameters, i.e., the weights w_{jk} in the output layer, the centers $\boldsymbol{c}_j$, and the spread parameter σ of the Gaussian function, which are clearly seen in Equation 1. One major challenge in the structural design of a powerful RBF is the determination of the RBF centers. The simplest way to determine the RBF

NN centers or clusters, $\boldsymbol{c}_j$, is by using randomly fixed clusters obtained through the information contained in the actual data. However, quantification of a sufficient number of centers that will give an adequate partition of the input space is difficult. Another method is to select a relatively large number of input vectors as the centers; however, since the major challenge in the RBF design parameter for higher recognition capability is choosing these centers, we use a *K-means* clustering algorithm in our method. The spread parameter σ is then set according to the simple heuristics relationship (such as in Haykin, 1996), which leaves the weight initialization problem, which is the main point of interest in this paper.

The weights of the BP NN are usually initialized with random numbers in the interval of -0.5 to 0.5, which are then modified by Nguyen-Widrow method (Nguyen & Widrow, 1990), which significantly improves the network training speed and reduces the computational cost. For every input-output data pair, the learning mechanism of the RBF NN is accomplished by adjusting the three sets of network parameters through supervised learning called the stochastic gradient-based method (Haykin, 1996). After calculating the error difference between the real output and the target, this value is propagated back to calculate and adjust the new weights in the output layer. In contrast to the learning mechanism in a BP NN, along with the calculation of new weights in the output layers of the RBF NN, both the position of the centers and the spread parameter for every center in the hidden layer are also changed accordingly.

In this paper, we have modified the RBF NN weight initialization by using an orthogonal least squared (OLS) method. In this type of weight initialization, the matrix weights of the RBF NN are derived by first calculating the orthogonal least square of the regression mapping between the output neurons and their target vectors. The use of the OLS weight initialization method allows batch processing of the learning process of the RBF NN, which reduces the computational learning cost of the networks when compared with that of the standard single vector learning mechanism. We have also modified the standard RBF NN by using a sigmoid function in the output neuron, as the addition of this function might results in a higher recognition capability of the networks. Consequently, the error function for updating the three set parameters based on a quadratic error function of the RBF NN should also be re-derived and modified.

The OLS method is actually derived from a linear regression problem, and as the mapping performance of the RBF NN can be represented as a regression model, the OLS method is utilized. In our proposed learning mechanism for the RBF NN, however, the OLS method is only applied for the initial neural network weights, since the learning mechanism of the three set parameters of the RBF NN used in the previously discussed method already shows a better performance. The relation between the target vectors of the RBF NN with its hidden neurons output times the weights, in the form of orthogonal least squared regression, can be written as:

$$\begin{bmatrix} G^1 \\ G^2 \\ \vdots \\ G^P \end{bmatrix} = \begin{bmatrix} \phi(x_1, c_1, \sigma_1) & \phi(x_1, c_2, \sigma_2) & \cdots & \phi(x_1, c_K, \sigma_K) \\ \phi(x_2, c_1, \sigma_1) & \phi(x_2, c_2, \sigma_2) & \cdots & \phi(x_2, c_K, \sigma_K) \\ \multicolumn{4}{c}{\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots} \\ \phi(x_P, c_1, \sigma_1) & \phi(x_2, c_2, \sigma_2) & \cdots & \phi(x_P, c_K, \sigma_K) \end{bmatrix} . \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_P \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ \vdots \\ e_P \end{bmatrix} \tag{2}$$

or

$$\boldsymbol{G} = \boldsymbol{\Phi w} + \boldsymbol{e} \tag{3}$$

where $\boldsymbol{G} \in \Re^{Px1}$ is the target vector of the network output, $\boldsymbol{\Phi} \in \Re^{PxK}$ can be regarded as a linear regression matrix with each column vector $\boldsymbol{\Phi} \in \Re^{Px1}$ being a regression vector or regressor, $\boldsymbol{w} \in \Re^{Kx1}$ is the vector of weights, and $\boldsymbol{e} \in \Re^{Kx1}$ is the vector of errors between the target vector and the actual network outputs. The use of all available input vectors as hidden neurons clearly shows that the total error of the RBF NN is nearly zero; however, the large size of the network will lead to a higher computational cost. The use of the OLS method for an initialization condition allows us to assume that the initial error could be eliminated, and if we suppose that $K < P$ is valid, then the least-squares solution yielding the initial weights is given by:

$$\hat{\boldsymbol{w}} = (\boldsymbol{\Phi}^{\boldsymbol{T}}\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^{\boldsymbol{T}}\boldsymbol{G} \tag{4}$$

The RBF-OLS NN learning is then performed by first using the defined initial weight $\hat{\boldsymbol{w}}$ derived from Equation 4. A feed-forward RBF OLS NN is then calculated by using Equation 1; however, we can used a batch processing mode. The matrix of errors between the output neurons and the target output is calculated in the same way as a normal RBF NN learning mechanism, and the network weights are updated, together with the other two parameters, using a gradient descent of quadratic error function, as shown in Equations 6, 7, and 8.

These two learning algorithms of the RBF NN are explained as follows, to provide a comparison of the characteristics of the developed method.

3.1. Weight Initialization of RBF-NN using a Nguyen-Widrow Method

Step 1. Initialize the centers using the defined number of clusters, the spread parameter, and the weights in the output layer using the Nguyen-Widrow method

Step 2. Present an input vector, and compute the network output

$$\begin{gathered} y_{ink}^{p} = \sum_{j=1}^{J} w_{jk}\phi_j\left(\| \boldsymbol{x}^p - \boldsymbol{c}_j \|\right) \\ y_k^p = \frac{1}{1 + e^{-y_{ink}^{p}}} \end{gathered} \tag{5}$$

Step 3. Update the three networks parameters according to

$$w_{jk}(t+1) = w_{jk}(t) + \alpha\delta_k\phi_j \| \boldsymbol{x}^p - \boldsymbol{c}_j \| \tag{6}$$

$$\boldsymbol{c}_j(t+1) = \boldsymbol{c}_j(t) + \beta\delta_k\phi_j \| \boldsymbol{x}^p - \boldsymbol{c}_j \| \sum_{k=1}^{K} w_{jk}\frac{\sum_n^N (x^p - c_j)}{\sigma^2} \tag{7}$$

$$\sigma(t+1) = \sigma(t) + \gamma\delta_k\phi_j \| \boldsymbol{x}^p - \boldsymbol{c}_j \| \sum_{k=1}^{K} w_{jk}\frac{\sum_n^N (x^p - c_j)^2}{\sigma^3} \tag{8}$$

where: $\delta_k = (D_k - y_{k)}y_k(1 - y_k)$, with$D_k$ the k-component of the target vector

α, β, γ : learning rate for the weight, the center, and the spread, respectively.

Step 4. Stop if the network has converged; else, go back to Step 2.

3.2. Weight Initialization of RBF-NN using an Orthogonal Least Squares Method

Step 1. Initialize the centers using the defined number of clusters, the spread parameter, and the weights in the output layer using OLS method, through:

a. Calculate matrix $\boldsymbol{\phi}$ in the form of Equation 2.

b. Calculate matrix $\boldsymbol{W}$ in the form of Equation 4.

Step 2. Present all $\boldsymbol{P}$ input vectors, and compute the network output, batch calculation of Equation 5.

Step 3. Update the three networks parameters in a batch calculation according to Equations 6, 7, and 8.

Step 4. Stop if the network has converged; else, go back to Step 2.

4. EXPERIMENTAL RESULTS AND DISCUSSION

The classification of pellet quality (*PQ*) can be formulated as a function of five parameters, i.e., *H*, *V*, *W*, *D*, and *TD*. The total data are divided into two datasets: a training dataset and a testing dataset. The training dataset is used to train the RBF NN, while the testing dataset is used for evaluating the model performance. In this study, the first dataset that contains 50% of the records is used as the training dataset, while the other 50% is used as the testing dataset. The second dataset contains 70% of the records used for the training dataset, while the remaining 30% is used as the testing dataset. The performance of the RBF NN models is evaluated in terms of three performance metrics, i.e., root mean square error (RMSE) for measuring the mean deviation (error) of the measured *PQ* values from the *PQ* values predicted by the model; mean absolute error (MAE) that measures how close the model predictions are to the eventual outcomes; and goodness of fit (R-squared), which generally takes values between *0* and *1*. When the R-squared approaches 1, the regression points tend to align more accurately along the model curve. These performance measures are mathematically described as follows (Alotaibi et al., 2007; Sutarya & Kusumoputro, 2011):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(PQ_t - PQ_y)^2} \tag{9}$$

$$MAE = \frac{1}{N}\sum_{1}^{N}\left|PQ_t - PQ_y\right| \tag{10}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(PQ_t - PQ_y)^2}{\sum_{i=1}^{N}(PQ_t - \overline{PQ}_y)^2} \tag{11}$$

where N is the length of the measured data matrix, PQ_y is the estimated value, PQ_t is the observed value, $\overline{PQ}_y$ is the average of the estimated values, and $\overline{PQ}_t$ is the average of the observed values

Table 1 shows a comparison of the recognition rate of the four types of RBF NN methods used in these experiments. The experiments are conducted ten times consecutively and the average values of the overall experiments are presented. The table clearly shows that, when the number of iterations is one, which means that the learning process is performed by only one feed forward calculation without further learning, the recognition rate for the testing dataset of the all conventional RBF NN, with or without the use of the sigmoid activation function on its output neurons, is 73.3%. This recognition rate is also the same when the learning mechanism of the conventional RBF NN is increased up to a thousand iterations.

Table 1 Comparative performance of the RBF NN within a five-dimensional problem space for a 50:50 percentage of the training-testing paradigm

		iterations		**RMSE**	**MAE**	**R^2**
		1	1000			
RBF	Train	86.7	86.7	0.271	0.191	0.699
	Test	73.3	73.3	0.341	0.240	0.477
RBF+ SIGMOID	Train	86.7	85.3	0.296	0.097	0.605
	Test	73.3	73.3	0.427	0.197	0.181
RBF-OLS	Train	86.7	93.3	0.196	0.127	0.826
	Test	73.3	89.3	0.308	0.185	0.573
RBF-OLS+ SIGMOID	Train	86.7	97.3	0.127	0.036	0.927
	Test	73.3	90.7	0.244	0.086	0.732

When using the RBF-OLS NN, however, the recognition rate results show different characteristics. Similar to the conventional RBF NN, the recognition rate for one iteration of the RBF-OLS NN, with or without the use of the sigmoid activation function at the output neurons, is 73.3%. However, when the learning mechanism of the RBF-OLS NN is increased up to a thousand iterations, the recognition rate is increased considerably up to 89.3% for the RBF-OLS NN without the sigmoid activation function, and 90.7% for the RBF-OLS NN with the sigmoid activation function.

Table 2 Comparative performance of the RBF NN within a three-dimensional problem space for a 70:30 percentage of the training-testing paradigm

		iterations		RMSE	MAE	R^2
		1	1000			
RBF	Train	82.9	82.9	0.294	0.204	0.611
	Test	77.8	77.8	0.203	0.093	0.566
RBF+ SIGMOID	Train	82.9	84.8	0.294	0.113	0.610
	Test	77.8	77.8	0.230	0.056	0.445
RBF-OLS	Train	82.9	92.4	0.179	0.119	0.857
	Test	77.8	88.9	0.222	0.115	0.778
RBF-OLS+ SIGMOID	Train	82.9	96.2	0.154	0.048	0.894
	Test	77.8	91.1	0.116	0.038	0.710

The analysis of the three performance characteristics also confirmed the superior performance of the RBF-OLS NN compared to the conventional RBF NN, especially when the iteration number was one thousand. As can be clearly seen in Table 1, the characteristics of the RBF-OLS NN outperformed the conventional RBF NN method, especially the RBF-OLS NN with the sigmoid activation function that shows the highest recognition rate, the lowest RMSE, lowest MAE, and highest R-squared parameters. Note that the recognition rate of the RBF-OLS NN is the same for all ten experiments, for both the training dataset and the testing dataset, confirming that the RBF-OLS NN performed with a very high stability in terms of its recognition rate calculation. The experimental results also show that the computational cost of the RBF-OLS NN is about 23 seconds, also lower when compared with the RBF NN, which is about 70 seconds for a thousand iterations.

Table 2 shows the experimental results for a comparison of the conventional RBF NN and the RBF-OLS NN, which are combined with or without a sigmoid activation function for the output neurons, using a training-testing paradigm of 70:30%. In general, the recognition rate for all the RBF NNs used with a training-testing paradigm of 70:30% shows a higher value when compared with the values from Table 1, which was conducted with a training-testing paradigm of 50:50%. Similar to the values shown in Table 1, the recognition rate for the conventional RBF NN is not affected by the use of a sigmoid activation function, or even by increasing the number of learning iterations. However, Table 2 clearly shows that the recognition rate of the RBF-OLS NN with a sigmoid activation function is higher than that obtained for the RBF-OLS NN without a sigmoid activation function.

As also confirmed by the results in the Table 1, Table 2 shows that the RBF-OLS NNs performed higher recognition rate compared with that of the conventional RBF NNs method. Table 2 also shows that the three parameter of the performance indicator values, i.e., RMSE, MAE and R^2 are in accordance with the results in the Table 1, showing the superiority of the RBF-OLS NNs. Further analysis, however, the R^2 values of the overall neural networks model used in these experiments, except for the RBF-OLS NN with a sigmoid activation function, are lower than 0.5, which means that the outputs of the neural networks model are not well aligned with the real testing dataset. In order to increase the R^2 value, a different strategy is developed as follows.

As already explained earlier, the recognition capability for pellet quality (*PQ*) classification depends on five parameters, i.e. *H*, *V*, *W*, *D* and *TD*, which means that the neural networks classifier should derive its decision line in a five dimensional problem space. Decreasing the problem space dimensions by looking for only orthogonal dimensions of the problem space will eventually lead to a robust decision line of the neural network classifier, thereby increasing the recognition capability of the neural networks.

The Karhunen-Loeve transformation (KLT) or a subspace transformation through a Principal Component Analysis method (PCA or SVD) (Fukunaga, 1990; Swets & Weng, 1996) is used to transform the original problem space into its optimal problem space. The purpose of the problem space transformation is to orthogonalize the intrinsic feature of the problem space, which usually reduces the original problem space and increases the separability characteristics of the classifier in the new problem space (eigenspace). Details of the K-L transformation are not explained in this paper, but can be seen in Kusumoputro et al. (2011).

Table 3 and Table 4 show the comparative performance of the RBF classifier used in this experiment after the problem space transformation is conducted from a five-dimensional problem space into a two-dimensional problem space, for two different training-testing paradigms, respectively.

Table 3 Comparative performance of the RBF NN within a three-dimensional problem space for a 50:50 percentage of the training-testing paradigm

		iterations		RMSE	MAE	R^2
		1	1000			
RBF	Train	97.3	97.3	0.190	0.130	0.837
	Test	96.0	96.0	0.212	0.136	0.798
RBF+	Train	97.3	85.3	0.166	0.036	0.876
SIGMOID	Test	96.0	96.0	0.166	0.031	0.876
RBF-OLS	Train	97.3	97.3	0.179	0.119	0.857
	Test	96.0	97.3	0.199	0.120	0.823
RBF-OLS+	Train	97.3	98.7	0.133	0.034	0.920
SIGMOID	Test	96.0	98.7	0.123	0.031	0.932

Table 4 Comparative performance of the RBF NN within a three-dimensional problem space for a 70:30 percentage of the training-testing paradigm

		iterations		RMSE	MAE	R^2
		1	1000			
RBF	Train	97.1	97.1	0.194	0.033	0.831
	Test	97.8	97.8	0.137	0.058	0.805
RBF+	Train	97.3	85.3	0.142	0.026	0.910
SIGMOID	Test	96.0	96.0	0.105	0.013	0.880
RBF-OLS	Train	97.3	97.3	0.188	0.023	0.841
	Test	96.0	97.3	0.132	0.052	0.818
RBF-OLS+	Train	97.3	98.7	0.170	0.044	0.831
SIGMOID	Test	96.0	98.7	0.104	0.019	0.887

These tables clearly show that the problem space transformation leads to a better value for the three parameters of the classifier characteristics, i.e. lower RMSE, lower MAE, and higher R^2 values. As the performance characteristics of the neural networks classifiers are clearly upgraded in both the training phase and the testing phase, the recognition rate of the neural networks classifiers are considerably increased in both training and testing phases. The increment in the recognition rate when using the conventional RBF NN, with or without a sigmoid activation function, indicates a considerable upgrading, with almost a 20% difference from that seen in the original problem space. For the RBF-OLS NNs, however, the increment in the recognition rate is considerably lower, perhaps because the recognition rate is already very high when using these classifiers in the original problem space.

5. CONCLUSION

An automatic assessment of the quality classification for green pellets is developed using computational intelligence methods. We have developed and utilized an RBF-OLS NN with a sigmoid activation function on its output neuron, and we analyzed the classifier characteristics through three parameters of an error-based system, i.e. RSME, MAE, and *R*-squared. The experimental results show that the RBF-OLS classifier system outperformed the conventional RBF NN method, especially when the problem space was in the original problem space with an intrinsic feature space that might not be orthogonal. Furthermore, a sigmoid activation function on its output neuron increased the recognition capability of the RBF-OLS classifier. The experiments also proved that transforming the original problem space into an optimized problem space through an eigen decomposition method significantly increased the recognition rate of the classifier, especially for the conventional RBF NN.

6. ACKNOWLEDGEMENT

The authors gratefully acknowledge the Ministry of Research and Higher Education for the Universitas Indonesia Cluster Research Funding 2015.

7. REFERENCES

Alotaibi, F.D., Abdennour, A., Ali, A.A., 2007. A Robust Prediction Model using ANFIS based on Recent TETRA Outdoor RF Measurements Conducted in Riyadh City-Saudi Arabia. *International Journal of Electronic Communication* (AEU), Volume X(x), pp. X–x

Batan, 2007. *Safety Analysis Report-IEBE*, Center of Nuclear Fuel Technology, no.doc. KK20J09003

Briyatmoko, B., Rachmawati, M., Yulianto, T., 2010. Current Status of R&D of Nuclear Fuel Elements for PWR in Indonesia. *In*: Advanced Fuel Pellet Materials and Fuel Rod Design for Water Cooled Reactors, IAEA-TTECDOC-1654, pp. 99–104

Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition.* Academic Press, San Diego, California, USA

Garcia, A., Luviano-Juarez, A., Chaires, A., Poznyak, A., Poznyak, T., 2011. Projectional Dynamic Neural Networks Identifier for Chaotic Systems: Applications to Chua Circuit. *International Journal of Artificial Intelligence*, Volume 6, S11, pp. 1–18

Haykin, S., 1996. *Adaptive Filter Theory*, 3rd ed., Prentice-Hall, Upper Saddle River, New Jersey, USA

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer Feedforward Networks are Universal Approximators, *Neural Networks*, Volume 2(5), pp. 359–366

Jayaraj, R.N., Ganguly, C., 2003. Recent Developments in Design and Manufacture of Uranium Dioxide Fuel Pellets for PHWRs in India. *In:* Proceedings of a Technical Committee Meeting, IAEA-TECDOC-1416, Brussels, Belgium, pp. 13–20

Jothiprakash, V., Magar, R.B., Kalkuthi, S., 2009. Rainfalls-runoffs Models using Adaptive Neuro-Fuzzy Inference System (ANFIS) for an Intermittent River. *International Journal of Artificial Intelligence,* Volume 3, A03, pp. 1–23

Kusumoputro, B., Faqih, A., Sutarya, D., Lina, 2013. Quality Classification of Green Pellet Nuclear Fuels using Radial Basis Function Neural Networks. *In:* IEEE Proceedings of the 12th International Conference of Machine Learning and Applications, Florida, USA, pp. 194–198

Kusumoputro, B., Lina, Kresnaraman, B., 2011. Improvement of Recognition Capability of Fuzzy-Neuro LVQ using Fuzzy Eigen Decomposition for Discriminating Three-Mixture Fragrances Odor. *Information Technology Journal*, Volume 10(12), pp. 2385–2391

Nguyen, D., Widrow, B., 1990. Improving the Learning Speed of the 2-Layer Neural Networks by Choosing Initial Values of Adaptive Weights. *In:* Proceedings of the IJJN, Volume 3, San Diego, California, USA, pp. 21–26

Pramanik, D., Ravindran, M., Rao, G.V.S.H, Jararaj, R.N., 2010. Innovative Process Techniques to Optimize Quality and Microstructure of UO_2 Fuel for PHWRs in India. *In:* Advanced Fuel Pellet Materials and Fuel Rod Design for Water Cooled Reactors, IAEA-TTECDOC-1654, pp. 13–34

Sarma, K.K., 2009. Neural Network Based Feature Extraction for Assamese Character and Numeral Recognition. *International Journal of Artificial Intelligence*, Volume 2, S09, pp. 37–56

Sutarya, D., Kusumoputro, B., 2011. Quality Classification of Uranium Dioxide Pellets for PWR Reactor using ANFIS. *IEEE Tencon* 2011, Bali, Indonesia, pp. 314–318

Swets, D.L., Weng, J., 1996. Using Discriminant Eigenfeatures for Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Volume 18, pp. 831–836