

PAPER DETAILS

TITLE: Reconstruction of the Taguchi Orthogonal Arrays with the Support Vector Machines Method

AUTHORS: Selçuk YAZAR

PAGES: 129-137

ORIGINAL PDF URL: <https://dergipark.org.tr/tr/download/article-file/1443852>

Reconstruction of the Taguchi Orthogonal Arrays with the Support Vector Machines Method

Selçuk Yazar


Abstract—Design of Experiment (DOE) is a widely used method for examining experiments especially in industrial production and robust design processes. This method is a set of statistical approaches in which mathematical models are developed through experimental testing to estimate possible outputs and given input values or parameters. The method aims to determine the main factors that affect the results with the smallest number of experimental studies. In this study, L16 (2^{15}) orthogonal array, which was used in the Taguchi parameter design was reconstructed with the Support Vector Machines learning model and the Pearson VII kernel function. With this model, array elements were successfully classified in 87.04%. The new and original array were compared and 3.8% difference was measured between their Signal to Noise (S / N) ratios in an exemplary experiment.

Index Terms— Machine Learning, Support Vector Machine, Taguchi Design, Design of Experiments, Robust Design.

I. INTRODUCTION

TODAY, IN many manufacturing applications, Taguchi's orthogonal array catalogs are used for industrial designs. Genichi Taguchi redesigned offline quality control methods [1], which were developed in Japan after World War II in the 1980s under the name of robust design in AT & T Bell laboratories. This method is generally called Taguchi orthogonal array design. For instance, in this method, determining the optimum levels of the nine control variables requires 6.000 possible combinations, while only 18 different controllable variables are adequate [2]. To give another example, let's assume that eight factors are affecting the experiment during the development of a robust model. If one of these factors affects two and seven of them at three levels, $2 \times 3^7 = 4.347$ experiments should be performed to reach optimum values. If there are 15 parameters per three levels in this instance, $3^{15} = 14.348.907$ distinct experiments are required. Taguchi's orthogonal array method is the solution in this type of experiment designs in which applying is practically difficult. This technique is used conventionally for industrial production areas.

SELÇUK YAZAR, is with Department of Computer Engineering Trakya University, Edirne, Turkey, (e-mail: selcukyazar@trakya.edu.tr).

 <https://orcid.org/0000-0001-6567-4995>

Manuscript received December 11, 2020; accepted March 29, 2021.
DOI: [10.17694/bajece.839449](https://doi.org/10.17694/bajece.839449)

Also, Taguchi designs use in Environmental Sciences [3, 4], Agricultural Sciences [5], Physics [6], Chemistry [7], Statistics [8], and Medicine [9] frequently. However, Genichi Taguchi did not share any information about the methods that are used to construct these arrays. Furthermore, it is difficult to distinguish the links between Taguchi's arrays and the published similar arrays elsewhere [10, 11].

In this study, we aimed to regenerate the L16 (2^{15}) Taguchi array using support vector machines methods, based on two-level L4, L8, L12 arrays which are found in the Taguchi arrays catalog. We compared the array that we obtained and the original array with a sample experiment. In addition, the variation of the factors are affecting the results is discussed in the sample experiment.

This paper is organized as follows: In Section 2 we present information about orthogonal arrays. In section 3 we briefly describe the support vector machines and the Pearson VII function. Also, we provide a description of the sequential minimal optimization algorithm that we used for classification. Finally, in section 4 we describe our proposal and discuss our results.

II. ORTHOGONAL ARRAYS AND EXPERIMENT DESIGN

Orthogonal arrays were first described by C.R.Rao as hypercube in 1946 [12] and introduced as statistical combinatorial arrangements methods. An orthogonal array consists of N rows and k columns, selected entries from a set of s symbols or levels defined as $S = \{0, 1, \dots, s-1\}$. The term "Level" in the array definition is used to indicate the level of factors and variables that have an impact on the test components in the experimental designs using these arrays. An $N \times k$ matrix A formed in this way is called an orthogonal array of level s and strength t . In such a notation s denotes the number of levels. If the number s is equal to 2, there are two levels in the array, 0 and 1.

The t value is defined in the interval $(0 \leq t \leq k)$. This value tells us the number of s^t possible t -tuples belonging to $N \times t$ subarrays that can be defined in array A . These t -tuples are also equal to the number of rows of the array. In addition, the value of λ is used to define the index of these arrays. The λ value indicates the number of repetitions of t -tuples up to s^t . When $\lambda = 1$, it is stated that the orthogonal array has the index unity. The orthogonal arrays are briefly denoted as $OA(N, k, t, s)$ or $OA(N, s^k, t)$. In summary, integers that are defined as N, k, s, t

are parameters of orthogonal arrays. The number of rows N is the number of runs indicating the number of experiments performed. The number of k columns refers to the combination of factors affecting the experiment. s is the number of levels, t is the strength. Let $OA(12, 11, 2, 2)$ be the 12 rows (number of runs), 11 columns (number of factors), 2 levels, and with strength 2 and of index unity orthogonal array. This array is shown in Table 1.

TABLE I: EXAMPLE OF A STANDARD 2 LEVELS ORTHOGONAL ARRAY.

0	0	0	0	0	0	0	0	0	0	0
1	1	1	0	1	1	0	1	0	0	0
0	1	1	1	0	1	1	0	1	0	0
0	0	1	1	1	0	1	1	0	1	0
0	0	0	1	1	1	0	1	1	0	1
1	0	0	0	1	1	1	0	1	1	0
0	1	0	0	0	1	1	1	0	1	1
1	0	1	0	0	0	1	1	1	0	1
1	1	0	1	0	0	0	1	1	1	0
0	1	1	0	1	0	0	0	1	1	1
1	0	1	1	0	1	0	0	0	1	1
1	1	0	1	1	0	1	0	0	0	1

For instance, let the first and last columns be selected of the exemplary orthogonal array described in Table 1. In this case, obtained new array shown in Table 2.

TABLE II: RANDOMLY SELECTED ARRAY COLUMNS

0	0
1	0
0	0
0	0
0	1
1	0
0	1
1	1
1	0
0	1
1	1
1	1

If pay attention to Table 2, four distinct rows can be seen. These are 0 0, 0 1, 1 0, 1 1 matrices in order and can be seen they repeat three times. 0 and 1 values in this array can be defined statistically as two different elements. For instance, consider that an experiment is conducted on food ingredients. Designers have standardized all other ingredients except sugar amount and fat type. The values in Table 2 could be changed, “sugary”, “sugarless”, “margarine”, “butter” respectively. It shows that there are two factors in this experiment and these factors have two levels. In this respect, this array is called a two-level array. 11 columns in the matrix, 11 different variables, 12 rows tell 12 different experiments can be done depending on these 11 variables.

Also, Mixed-level orthogonal arrays are defined as well as sequences with two levels. In cases where factors affecting an experiment have different levels, the orthogonal arrays that are used may expand accordingly. In mixed-level orthogonal arrays, there are more than one S sets where the levels are obtained. In this case, the mixed-level arrays are

shown as $OA(N, s_1^{k_1}, s_2^{k_2}, \dots, s_n^{k_n}, t)$. The first k_1 column in the array is obtained from the S_1 set, the next k_2 column from the S_2 set. However, there is no index definition in mixed-level arrays.

The algorithms for constructing standard orthogonal arrays have been clearly explained mathematically over time. Although Taguchi arrays are formed from existing arrays, there is no known method for all. Taguchi has a total of 21 orthogonal arrays of 2, 3, and 4 levels, in order. Unlike existing orthogonal notations, Taguchi arrays are denoted like below.

$$L(\text{number of experiments})(\text{Level}^{\text{Factor}})$$

Here, for example, the array denoted as $L4(2^3)$, 4 is the numbers of experiments (runs), 3 factors with 2 levels (variable) for each factor. Similarly, mixed-level sequences are defined in the Taguchi catalog. For example, an $L8(2^3 4^1)$ means that the design has 8 runs, 3 factors with 2 levels, and 4 factors with 1 level. $L4(2^3)$ and $L8(2^3 4^1)$ are shown in Table 3.

TABLE III: (A) $L4(2^3)$ 2-LEVEL ORTHOGONAL ARRAY, (B) $L8(2^3 4^1)$ MIXED-LEVEL ORTHOGONAL ARRAY

	1	2	3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

	1	2	3	4	5
1	1	1	1	1	1
2	1	2	2	2	2
3	2	1	1	2	2
4	2	2	2	1	1
5	3	1	3	1	2
6	3	2	1	2	1
7	4	1	2	2	1
8	4	2	1	1	2

In Taguchi designs, for the analysis of the results are obtained from the experiments, the performance statistics method called Signal to Noise (S/N) ratio is used. Performance statistics are used to measure the effects of uncontrollable factors (noise factors) in the experiment. Noise factors (uncontrollable factors) cause deviation in the values that are obtained from the results of the experiment. There are three S/N ratios in the Taguchi design shown in Table 4.

TABLE IV: SIGNAL-TO-NOISE RATIO CALCULATION METHODS

S/N Characteristic	S/N Ratio	Comments
Smaller is better	$-10 \log(\frac{1}{n} \sum y_i^2)$	n = Number of responses in the factor level combination. y_i = Responses for the given factor level combination.
Nominal is best	$-10 \log(\frac{\bar{y}}{s^2})$	\bar{y} = Mean of responses for the given factor level combination s = Standard deviation of the responses for the given factor level combination
Larger is better	$-10 \log(\frac{1}{n} \sum \frac{1}{y_i^2})$	y_i = Responses for the given factor level combination n = Number of responses in the factor level combination.

As the ratio S/N increases, the result variance decreases around the target value. While determining the factor level is preferred for effective factors, the value with the highest S/N ratio is taken into consideration in experiment design.

III. SUPPORT VECTOR MACHINES AND PEARSON VII KERNEL

Support vector machines (SVM) are located in supervised learning in the machine learning paradigm. It is used for case classification and pattern recognition in many different research disciplines such as medicine [13] and signal processing [14]. It is also an effective method to use decision trees with SVM to solve multi-class problems. Decision trees are preferred to determine measurement performances and make predictions [15]. The algorithms are used in SVM applications are very successful in classification problems. They are used to classify data sets that cannot be separated linearly. In summary, the algorithms are used herein look for a linearly separable hyperplane or a decision boundary that separates members of one class from another. If this decision boundary is found to be immensely successful, the desired classification result is achieved. An SVM classification model requires two main components: the support vector itself and the optimal decision limit.

Assuming that there is a set of two classes that can be easily separated from each other, there can be an infinite number of linear planes separating these two classes. If any i indexed point represented as (X_i, Y_i) has definite values such as +1 or -1, it is necessary to identify the planes separating these points

from each other and to determine which of these planes makes the best classification. The example classification result is shown in Fig. 1.

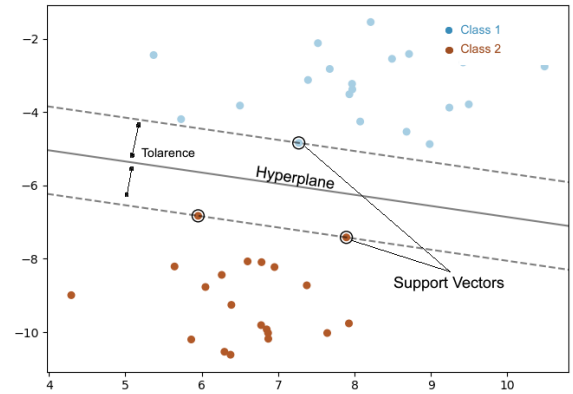


Fig. 1: Example of two-dimensional support vector machines classification.

When the values in classes 1 and 2 are displaced over time, there is a need to reconstruct the obtained planes to determine the optimum state and to rebuild the model. SVM algorithms are used for this purpose. Two classification methods are used in the SVM. These methods are called C-SVM and nu-SVM.

C-SVM is an error function is used conventionally to improve the compatibility of SVM. In the C-SVM model, the C value is used to systematically control the tolerance of discrete values. The coefficient C is an empirical parameter that works with grid search management. C-SVM function shown in equation 1, where w is the vector indicates the coefficients and φ_i is the slack variable and is also known as the training error for the training vector.

$$\tau(w, \varphi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \varphi_i \quad (1)$$

nu-SVM models have been developed as an alternative to the C parameter. In the solution of nonlinear problems, the C value is used as a weight parameter to measure learning errors and tolerance of plane and support vectors in the classification process. In equation 1 the parameter C is replaced by ϑ parameter. The parameter ϑ in the nu-SVM model represents the fraction of the support vectors expected in the solution of the problem. Thus, for any given value of $(0, 1]$, the shape of the classifier can be predetermined. nu-SVM classifier shown in equation 2.

$$\tau(w, \varphi, \rho) = \frac{1}{2} \|w\|^2 - \vartheta \rho + \frac{1}{N} \sum_{i=1}^N \varphi_i \quad (2)$$

subject to $\varphi_i \geq 0, \rho \geq 0$

In n-dimensional space, a hyperplane is an n-1-dimensional flat subspace. As it can be seen in Fig. 1, the hyperplane is a one-dimensional line, which is found as a separator in 2-dimensional space. The hyperplane of the classes distributed in 3-dimensional space will be a 2-dimensional plane. If expressed mathematically; In a space with n dimensions, a separating

hyperplane will be the linear combination of dimensions with a sum of 0.

$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n = 0 \quad (3)$$

The value θ_0 in equation 3 is often referred as a deviation. If the value θ_0 equals zero, the hyperplane will be at the origin point.

However, most classification problems cannot be solved by a linear function. There is various type of functions to obtain the hyperplane between the two classes and these functions are called the kernel. Besides the linear kernel function, polynomial, radial based, and sigmoid functions are also used. The kernel functions are expressed as $K(X_i, X_j)$. In the kernel functions, the data set classes are separated by using the C and ϑ parameters [16, 17].

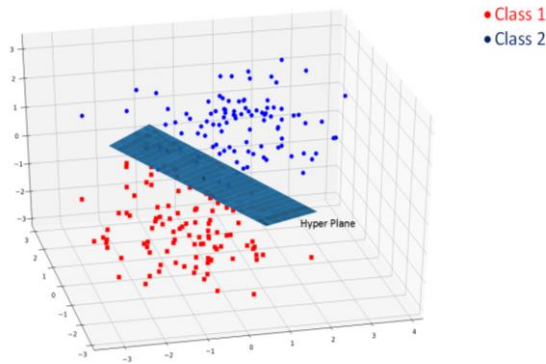


Fig. 2: Example of three-dimensional support vector machine classification.

As shown in Fig. 2, when classification solutions cannot always be easily solved by a linear function, alternative solutions can be found in spaces with increased dimensions. The kernel functions and dependent parameters should be determined according to the solution. In addition to linear solutions, different kernel functions should be used to separate and classify data sets belonging to multiple and nested classes.

A. Pearson VII kernel function

The Pearson VII function was used for classification that was developed in 1895 for X-ray diffraction scanning and for more accurate estimation of Gauss and Lorentz graphs with the parameter ω . Thus, these graphs, which are formed by the control of ω, σ parameters, have behaved similarly to the sigmoid function which is frequently used in artificial neural network research [18]. The kernel functions which are used in support vector machines are symmetric and semi-defined positive. In general, any kernel function fulfilling the Mercer's theorem conditions belongs to the class of valid kernel functions. The Mercer's theorem implies that a valid kernel function must be symmetric [19]. The resulting symmetrical matrix representing the kernel matrix must be positive semi-definite. The Pearson VII function is as follows:

$$K(x_i, x_j) = \frac{1}{\left[1 + \left(\frac{2 \sqrt{\|x_i - x_j\|^2} \sqrt{2^{(1/\omega)} - 1}}{\sigma} \right)^2 \right]^{2\omega}} \quad (4)$$

where x_i and x_j are two vector arguments. The single variable in the original version of the function is replaced by two vector arguments so that the Euclidean distance is measured between them. If one of the symmetrical or positive semi-definite conditions is met, then the conditions of the Mercer's are supplied. As a consequence, the PUK function satisfies all these requirements.

B. Sequential minimal optimization-SMO

The SMO algorithm is used to train support vector machines classifier functions. The SMO is developed to solve the known optimization problem in quadratic programming [20]. The optimization problem is shown as follows,

$$\max_{\alpha} \psi(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j k(x_i, x_j) \alpha_i \alpha_j \quad (5)$$

Equation 5 is dependent on the terms $\sum_{i=1}^N y_i \alpha_i = 0, 0 \leq \alpha_i \leq C$ and $i = 1, \dots, n$. Here, x_i represents the training example, $y_i \in \{-1, +1\}$ the dependent target value, α_i represents the Lagrange multiplier and C represents an SVM hyperparameter.

This method solves the main optimization problem of SVM by accepting two Lagrange multipliers $\{\alpha_1, \alpha_2\}$ as a starting value of $\alpha_1^{old} = \alpha_2^{old} = 0$ in a previous set of solutions $\{\alpha_1^{old}, \alpha_2^{old}, \alpha_3^{old}, \dots, \alpha_n^{old}\}$ for a classification problem.

In equation 5, given $t = y_1 y_2$, $y_1 \alpha_1 + y_2 \alpha_2 = y_1 \alpha_1^{old} + y_2 \alpha_2^{old} = \text{constant}$ and $\alpha_1 = \sigma - t \alpha_2$ values are replaced and also eliminating α_2 via the first and second derivatives;

$$\alpha_j^{new} = \alpha_j^{old} + \frac{y_j (E_j^{old} - E_i^{old})}{\eta} \quad (6)$$

The α_j^{new} value is obtained in equation 6. Here E_i prediction error and the value η is defined as $2k_{ij} - k_{ii} - k_{jj}$. The value E_i is the error that the i th training example and $E_i = (u_i - y_i)$ where u_i is the plane of the classification.

The main problem encountered in the training of data sets in the SVM method is that time performance in the training sets into chunks is poor. This is called a constrained quadratic programming problem in the literature. The SMO algorithm greatly improves the training performance of sparse data sets in these classification tasks. Although SMO offers a fast solution for sparse datasets, especially for linear SVMs, this algorithm can be extremely slow in non-sparse datasets. In this paper, we did not have a performance problem as we are using a small dataset and a nonlinear kernel function.

IV. CLASSIFICATION RESULTS AND CONCLUSION

In this study, in order to reconstruct the L16 (2^{15}) Taguchi array, we formed the classification model by using the support vector machine model which belongs to L4 (2^3), L8 (2^7), L8 (2^4), and L12 (2^{11}) 2-level arrays [21]. During the creation of the model, we also tested with various kernel functions that were used for SVM. Moreover, we achieved the highest performance classification with the Pearson VII kernel (PUK)

function that was used with Sequential Minimal Optimization algorithm.

Firstly, we created the model for the L16 2-level orthogonal array with the PUK kernel. At the time of the model creation, the maximum successful classification results were obtained by changing the classifier parameters C, γ , and Pearson VII kernel function parameters ω (ω), σ (σ) is shown in Table 5.

TABLE V
COMPARISON OF CLASSIFICATION RESULTS WITH VARIOUS
KERNEL FUNCTIONS.

Method	Sigmoid	RBF	Linear	Polynomial	PUK
C-SVM	%54,4	%57,24	%47,03	%42	%66,4 (Same with SMO)
nu-SVM	%46,4	%67,2	%54,7	%47	%87,04 (SMO)

To classify the array data, we prepared a data set using the matrix elements of other 2-level orthogonal arrays. The confusion matrix of the results we obtained in Table 5 is shown in Fig. 3. As seen in Fig 3, most of the matrix elements are classified correctly thanks to the kernel function we use.

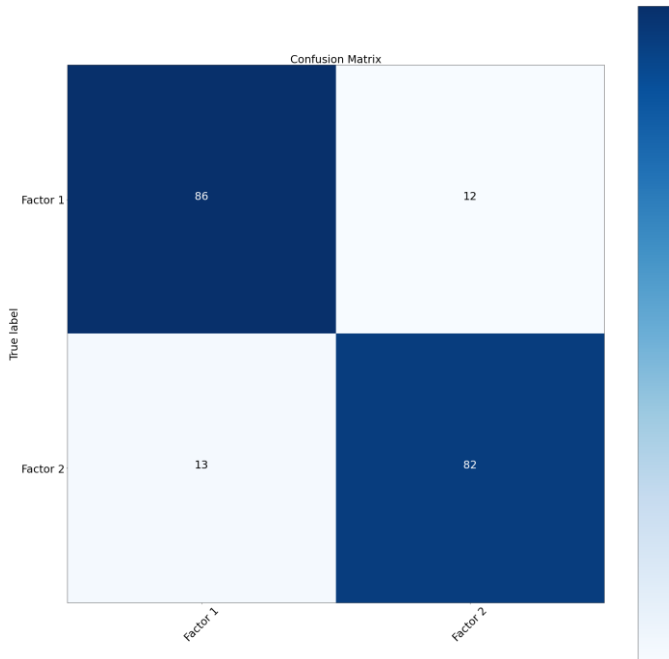


Fig. 3: The confusion matrix of the classification results we get from the 2-level arrays with nu-SVM and PUK kernel (SMO).

During the study, we observed that nonlinear core functions had higher performance in the classification of orthogonal arrays. The performance of the RBF and PUK cores is noticeably better than the other kernels, especially while classifying in 2-level classes. The output in the two-dimensional space obtained during the PUK classification process is shown in Fig. 4.

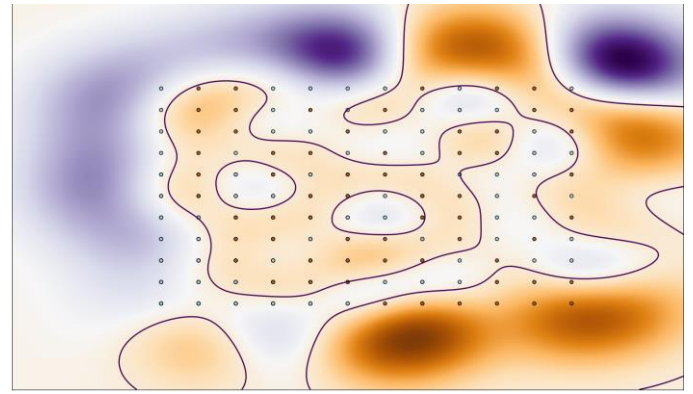


Fig. 4: Classification result with nu-SVM with Pearson kernel function without SMO. ($\gamma=0.5$, $\sigma(\sigma)=0.001$, $\omega(\omega)=0.001$, these values are obtained with experimentally for best results)

In this study, we have used the sequential minimal optimization algorithm for the values in the classification of the support vector machine. We achieved higher classification results with the SMO rather than the model that we applied without this algorithm. The plot of the classification results using the SMO algorithm is shown in Fig. 5.

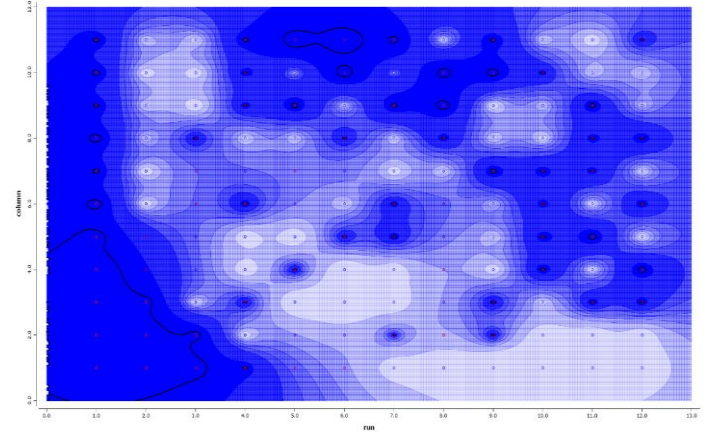


Fig. 5: nu-SVM classification plot, Pearson kernel function with SMO. ($\gamma=0.5$, $\sigma(\sigma)=0.001$, $\omega(\omega)=0.001$, these values are obtained with experimentally for best results)

We created the plot of the solution function and examined this graph in three dimensions to see if it is a sufficient hyperplane to differentiate the basic array values that we used and obtained during classification.

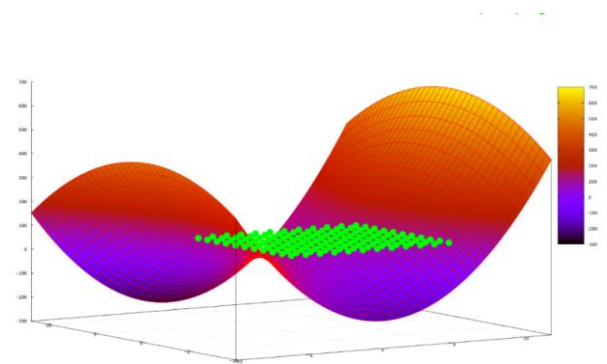


Fig. 6: Result function of our classification model.

Fig. 6 shows the values of the 2-level class shown in green, and the solution function of the classification with the highest performance is shown as a convex curve.

As a result of the most successful classification, we obtained the new orthogonal array is shown in Table 6. The red variables in Table 6 show values that differ from the original array. When we examined the distribution of elements in the array, we found it had a 40% difference from the original array.

Besides, the Taguchi arrays catalog contains mixed-level arrays as well as 2-level arrays which are defined in section 2.

We applied the model that is obtained for the L16 (2^{15}) array to find the L16 ($2^3 4^4$) mixed-level orthogonal array. To find the array L16 ($2^3 4^4$), we classified the arrays L16 ($2^6 4^3$), L16 ($2^9 4^2$), L16 ($2^{12} 4^1$), L8 ($2^4 4^1$), but we could not achieve satisfactory results. These results are shown in Table 7.

TABLE VI
NEW ARRAY CONSTRUCTED BY CLASSIFYING WITH nu-SVM(SMO).

Factors Experiment Number (runs)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	2	2	2	2	2	2	1	1	1	1
3	1	1	2	1	2	2	2	1	2	2	2	1	1	1	1
4	1	2	1	2	2	1	2	2	1	1	1	1	1	1	1
5	1	2	2	1	2	2	2	2	1	2	1	1	1	1	1
6	2	2	2	2	1	2	2	1	2	1	1	1	1	1	1
7	2	1	2	2	1	1	2	2	1	2	1	1	1	1	1
8	2	2	2	2	2	2	2	1	1	1	2	1	1	1	1
9	2	1	1	2	2	2	1	2	2	1	1	1	1	1	1
10	2	2	2	1	1	1	1	2	2	1	2	2	1	1	1
11	2	2	1	2	1	2	1	1	1	2	2	2	1	1	1
12	2	2	1	1	2	1	2	1	2	2	1	1	1	1	1
13	2	2	1	1	1	1	1	1	2	2	1	1	1	1	1
14	2	2	2	1	1	1	1	2	2	2	2	1	1	1	1
15	2	2	2	2	2	1	1	2	2	2	1	1	1	1	1
16	2	2	2	2	2	2	1	1	2	1	1	1	1	1	1

The reason why our method fails on mixed-level arrays is the SMO algorithm is used to solve an optimization problem with two variables. According to the results in Table 7, we saw that the algorithm we use in the classification method is not suitable for different factors and levels in mixed-level arrays.

On the other hand, the new array we obtained was not orthogonal. Therefore we could not fully analyze it in accordance with the principles of experimental design. However, in non-orthogonal variance analysis, we examined the S/N ratio values using the ANOVA (Analysis Of Variance) method. These results are shown in Table 8 and Table 9.

TABLE VII
CLASSIFICATION RESULTS FOR MIXED-LEVEL ARRAYS USING
THE LAST SUCCESSFUL MACHINE LEARNING MODEL

Method	Sigmoid	RBF	Linear	Polynomial	PUK
C-SVM	%42,2	%47,2	%41,6	%32	%36,4 (Same with SMO)
nu-SVM	%37,6	%37,2	%44	%47	%43,61(SMO)

TABLE VIII
STANDARD L16 TAGUCHI 2-LEVELS ARRAY SIGNAL TO NOISE RATIOS / LARGER IS BETTER

Factor	A	B	C	D	E	F	G	H	J	K	L	M	N	O	P
1	3,763	3,763	4,515	1,505	2,258	3,763	3,010	3,763	3,010	3,010	2,258	2,258	3,010	3,010	2,258
2	2,258	2,258	1,505	4,515	3,763	2,258	3,010	2,258	3,010	3,010	3,763	3,763	3,010	3,010	3,763
Delta	1,505	1,505	3,010	3,010	1,505	1,505	0,000	1,505	0,000	0,000	1,505	1,505	0,000	0,000	1,505
Rank	6,5	6,5	1,5	1,5	6,5	6,5	13	6,5	13	13	6,5	6,5	13	13	6,5

TABLE IX
RECONSTRUCTED TAGUCHI 2-LEVELS ARRAY WITH nu-SVM(SMO), SIGNAL TO NOISE RATIOS / LARGER IS BETTER

Factor	A	B	C	D	E	F	G	H	J	K	L	M	N	O	P
1	3,612	2,736	2,006	2,257	2,257	3,010	3,762	3,762	2,580	3,344	2,408	3,440	3,010	3,010	3,010
2	2,734	3,612	4,300	3,762	3,762	3,010	2,257	2,257	3,344	2,580	4,013	0,000	?	?	?
Delta	0,875	0,875	2,293	1,505	1,505	0,000	1,505	1,505	0,764	0,764	1,605	3,440	0,000	0,000	0,000
Rank	8,5	8,5	2	5,5	5,5	13,5	5,5	5,5	10,5	10,5	3	1	13,5	13,5	13,5

We examined the parameters of the sample experiment (Drought survey) shown in Table 8 for the reconstructed array. The important factor for the experiment here is, of course, which varies depending on the type of experiment, the characteristics of the components in the subject matter, and which change has a greater effect on the outcome of the experiment or process. Without changing the value and factor information in the original array, our results are obtained in the new array are shown in Table 9.

In addition to the results in Table 9, when we examined the data averages related to signal/noise ratios, we also observed that inactive parameters in the standard orthogonal array can be effective in the newly constructed array, taking into account the order of parameters. For instance, in the sample experiment in this study, the effect of the North-South aspect parameter (factor J), on the result of the newly constructed array is shown in Figs 7 and 8.

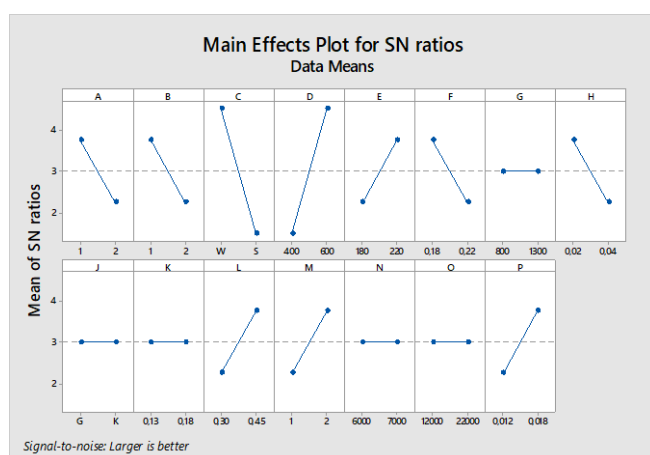


Fig. 7: Effects of standard L16 2-levels Taguchi orthogonal array parameters S/N ratios.

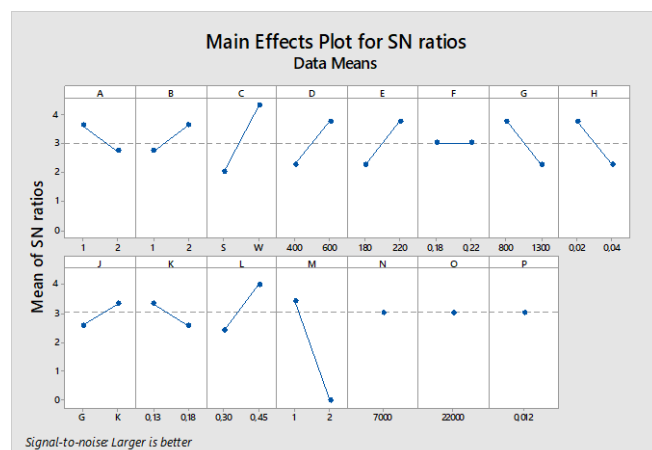


Fig. 8: Effects of new-generated 2-levels Taguchi orthogonal array parameters S/N ratios.

Another effect can be seen in Figures 7 and 8, in which we could not examine the effect of the last three parameters (N, O, P) on the experiment depending on the success of the classification. A better kernel function is used for the classification will eliminate the failure here. However, when these parameters are specific to the experiment are evaluated, it is seen if the effect of the results is low, and the classification is sufficient rather than unsuccessful. As a result of the classification we made under normal conditions, the deterioration of orthogonality seems to prevent the parameter effect research. Comparisons of our results are shown in Table 10.

TABLE X
COMPARISONS OF PREVIOUS STUDIES AND THE PROPOSED METHODS.

Author(s)	Array Size	Method	Intended for
Kenny Q. et al[22]	12	Combinatorial isomorphism.	New arrays generation
Sorana D. Bolboacă, Lorentz Jäntschi[23]	16	Computational algorithm	New arrays generation
H. Bayrak and A. Alhan [24]	8,9	Projective geometry algorithm	Array classification
Our Proposal	16	SVM with SMO	Array classification and reconstruction

As in the studies we have compared in Table 10, orthogonal array researches are particularly focused on creating new variations of 2, 3, 4, and mixed-level arrays with mathematical approaches. It is envisaged to develop and simplify the experimental designs with the arrays to be found in this way. Thus, a more useful optimization of production designs can be achieved using multiple factor levels and minimal experimental studies. In this paper, we used a novel approach that includes machine learning methods to obtain existing 2-level and mixed-level orthogonal arrays by classifying existing arrays. However, the model we obtained more successfully in fixed arrays, did not make the same performance in mixed-level arrays.

The effects of column values are more important in optimal design approaches in the non-orthogonal row-column structure [25]. It is known that non-orthogonal approaches are not suitable for optimal design methods. In addition, in some of the studies that inspired this study, artificial neural network training methods [26, 27], compared with the Taguchi method give more realistic results than orthogonal arrays. In this study, depending on the type of experiment and the relationship between physically investigated values, we investigated whether a Taguchi array can be obtained to approach the results of artificial neural networks. As shown in Table 9, we could not obtain the second level values of the factors in the last three columns. The effects of these columns need to be examined in more depth. Considering the order of factors affecting the experiment, forming an orthogonal array can give more accurate results.

Nowadays, as the usage areas of artificial intelligence are increasing, it is possible to improve the robust design approach, which is an important value in product design with machine learning methods. This requirement in production activities is very important in terms of more efficient use of resources.

REFERENCES

- [1] G. Taguchi and S. Konishi, *Taguchi Methods: Orthogonal Arrays and Linear Graphs. Tools for Quality Engineering*. Dearborn: MI: American Supplier Institute, 1987.
- [2] L. E. Katz and M. S. Phadke, "Macro-Quality with Micro-Money," in *Quality Control, Robust Design, and the Taguchi Method*, K. Dehnad, Ed. Boston, MA: Springer US, 1989, pp. 23-30.
- [3] P. T. Dhorabe, D. H. Lataye, A. R. Tenpe, and R. S. Ingole, "Adsorption of p-nitrophenol onto acacia glauca saw dust and waste orange peels

- activated carbon: application of Taguchi's design of experiment," *SN Applied Sciences*, journal article vol. 1, no. 3, p. 250, February 20 2019.
- [4] M. Rahmani, M. Kaykhaii, and M. Sasaki, "Application of Taguchi L16 design method for comparative study of ability of 3A zeolite in removal of Rhodamine B and Malachite green from environmental water samples," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 188, pp. 164-169, 2018/01/05/ 2018.
- [5] S. Sadr, V. Mozafari, H. Shirani, H. Alaei, A. Tajabadi Pour, and A. Rajabi Behjat, "Control of pistachio endocarp lesion by optimizing the concentration of some nutrients using taguchi method," *Scientia Horticulturae*, vol. 256, p. 108575, 2019/10/15/ 2019.
- [6] H.-A. Mehedi et al., "Synthesis of graphene by cobalt-catalyzed decomposition of methane in plasma-enhanced CVD: Optimization of experimental parameters with Taguchi method," *Journal of Applied Physics*, vol. 120, no. 6, p. 065304, 2016.
- [7] L. Zhang et al., "Taguchi method assisted multiple effects optimization on optical and luminescence performance of Ce: YAG transparent ceramics for high power white LEDs," *Journal of Materials Chemistry C*, 10.1039/C9TC03916C 2019.
- [8] R. Romero-Villafranca, L. Zúñica, and R. Romero-Zúñica, "Ds-optimal experimental plans for robust parameter design," *Journal of Statistical Planning and Inference*, vol. 137, no. 4, pp. 1488-1495, 2007/04/01/ 2007.
- [9] E. Y. Ng and W. K. Ng, "Parametric study of the biopotential equation for breast tumour identification using ANOVA and Taguchi method," (in eng), no. 0140-0118 (Print).
- [10] R. N. Kacker, E. S. Lagergren, and J. J. Filliben, "Taguchi's Orthogonal Arrays Are Classical Designs of Experiments," *Journal of research of the National Institute of Standards and Technology*, vol. 96, no. 5, pp. 577-591, Sep-Oct 1991.
- [11] P. J. Rahul Davis, "Application of Taguchi-Based Design of Experiments for Industrial Chemical Processes," in *Statistical Approaches With Emphasis on Design of Experiments Applied to Chemical Processes*, 2017.
- [12] C. R. Rao, "Hypercube of Strength "d" leading to Confounded Designs in Factorial Experiments," *Bull. Calcutta Math. Soc.*, vol. 38, pp. 67-68, 1946.
- [13] O. D. OLANLOYE, O. OLASUNKANMI, and O. E. ODUNTAN, "Comparison of Support Vector Machine Models in the Classification of Susceptibility to Schistosomiasis," *Balkan Journal of Electrical and Computer Engineering*, vol. 8, no. 3, pp. 266-271.
- [14] Y. Kaya, "Classification of PVC beat in ECG using basic temporal features," *Balkan Journal of Electrical and Computer Engineering*, vol. 6, no. 2, pp. 10-4, 2018.
- [15] E. Aydemir and F. Kaysi, "Development of Short Forms of Scales with Decision Tree Algorithms," *Balkan Journal of Electrical and Computer Engineering*, vol. 8, no. 4, pp. 285-290, 2020.
- [16] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," presented at the Proceedings of the fifth annual workshop on Computational learning theory, Pittsburgh, Pennsylvania, USA, 1992.
- [17] B. Scholkopf and A. Smola., *Learning with kernels*. M.I.T. Press, 2001.
- [18] B. Üstün, W. J. Melssen, and L. M. C. Buydens, "Facilitating the application of Support Vector Regression by using a universal Pearson VII function based kernel," *Chemometrics and Intelligent Laboratory Systems*, vol. 81, no. 1, pp. 29-40, 2006/03/01/ 2006.
- [19] J. Mercer, "Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations," *Philosophical Transactions of the Royal Society of London Series A*, vol. 209, p. 415, January 01, 1909 1909.
- [20] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," 1998.
- [21] Catalogue of Taguchi designs. Available: <https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/doe/supporting-topics/taguchi-designs/catalogue-of-taguchi-designs/>
- [22] K. Q. Ye, D. Park, W. Li, and A. M. Dean, "Construction and classification of orthogonal arrays with small numbers of runs," *Statistics and Applications*, vol. 6, no. 1, pp. 5-15, 2008.
- [23] S. D. Bolboacă and L. Jäntschi, "Design of experiments: Useful orthogonal arrays for number of experiments from 4 to 16," *Entropy*, vol. 9, no. 4, pp. 198-232, 2007.

- [24] H. Bayrak and A. Alhan, "On the construction of orthogonal arrays," Hacettepe Journal of Mathematics and Statistics, vol. 31, pp. 45-51, 2002.
- [25] J. Kunert, "A note on optimal designs with a non-orthogonal row—column-structure," Journal of Statistical Planning and Inference, vol. 37, no. 2, pp. 265-270, 1993/11/01/ 1993.
- [26] I. H. Kurt, M. Oduncuoglu, F. N. Yilmaz, E. Ergul, and R. Asmatulu, "A Comparative Study on the Effect of Welding Parameters of Austenitic Stainless Steels Using Artificial Neural Network and Taguchi Approaches with ANOVA Analysis," Metals, vol. 8, no. 5, 2018.
- [27] A. Alizadeh and H. Omrani, "An integrated multi response Taguchi-neural network- robust data envelopment analysis model for CO2 laser cutting," Measurement, vol. 131, pp. 69-78, 2019/01/01/ 2019.

BIOGRAPHIES



SELÇUK YAZAR received the B.S. degree in engineering physics from the Ankara University in 1997.

From 1998 to 2010, he worked as DevOps engineer and software developer. He received the M.S. and Ph.D. degrees in computer engineering from the Trakya University. His research interest include

Internet of Things, Deep Learning and Virtualization.